STRONG CONVEX RELAXATIONS FOR QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMS

A Dissertation
Presented to
The Academic Faculty

By

Asteroide Santana

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial & Systems Engineering

Georgia Institute of Technology

May 2019

STRONG CONVEX RELAXATIONS FOR QUADRATICALLY CONSTRAINED QUADRATIC PROGRAMS

Approved by:

Assoc. Professor Santanu S. Dey, Advisor
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Institute Professor George Nemhauser
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Assoc. Professor X. Andy Sun
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Professor Natashia Boland
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Assoc. Professor Yang Wang
School of Civil and Environmental Engineering & School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: March 7, 2019

“Whether you think you can

or you think you can’t

either way you are right.”

Henry Ford

ACKNOWLEDGEMENTS

In practical terms, this thesis is a compilation of several scientific contributions in the field of mathematical optimization. However, my perception of the document goes far beyond that simple description. Ultimately, this thesis represents the support, dedication, and time invested by many people–far too many to name–towards my development. Therefore, it is not just a document, but is the consummation of my Ph.D. degree. Or, more aptly, *this thesis is a celebration, together with all who participated, of my academic and personal growth over the last four and a half years*. In that context, I am thankful beyond words to the following:

My advisor, Professor Santanu Dey, for the countless direct and indirect contributions you made towards my success. Most of my notable achievements over the last few years resulted from your guidance. Besides teaching me how to do research and write papers, you helped me to be more professional, aim for the highest standards, understand the importance of seeking the big picture, set priorities, and develop many other skills now part of my personality. I acknowledge your financial support, which significantly improved the quality of life of my family. I am thankful also for your patience, when I made mistakes or had computational bugs, and for your willingness to shift research direction to meet my goal of working closer to the application. During the nearly four years we have worked together, you always offered encouragement, guiding me to the edge of my productivity, yet respected my limitations, motivations, and personal life. Over time, we exchanged feedback, improved communication of our expectations, and built a collaborative relationship, for which I am deeply grateful.

Burak Kocuk for your contributions to our research, your kindness, all the technical support, and sharing with me your inspirational way of thinking and solving problems. It was a honor working with you.

Professor Yang Wang for introducing us to the Finite Element Updating Model Problem and sharing with us your knowledge and insights on this application. Thanks also to Xinjun Dong for providing the data and for our insightful discussions.

The committee members, Professor Natashia Boland, Professor George Nemhauser, Professor Andy Sun, and Professor Yang Wang, and to the anonymous referees who reviewed the papers, for your precious time. Your inquiries and suggestions significantly improved the quality of the final document.

May and Eric from the ISyE helpdesk for the technical support you have provided. You always replied promptly and professionally to my requests and ensured that I kept going with my computational experiments.

Professor Alan Erera for trusting my capabilities and accepting me into the program. And Amanda Ford for helping me remain in good standing with respect to bureaucratic aspects of Ph.D. life. You always reminded me of important deadlines and quickly replied to emails.

Professor Maicon Marques Alves, my M.S. thesis advisor, for believing, even before I could, in my potential to succeed in this program. I cannot imagine how different my life would have been had you not encouraged me to pursue my Ph.D. degree in the United States. Moreover, you provided very much appreciated emotional support throughout the program. Your passion for science, and our conversations before I departed for the United States back in 2014, had a profound positive impact on my career and changed the destiny of my entire family.

Professor Renato Monteiro for all your support. I would certainly not have come to Georgia Tech had you not agreed to be my academic advisor. It was a great honor being guided by you during my first year in the program. I am very thankful for your financial support through the first year of the program, which significantly improved the quality of life of my wife and I. The precious time invested in teaching and sharing your wonderful understanding of the Nesterov's Method is greatly appreciated. Finally, I am most thankful for your understanding and willingness to allow me to pursue new research opportunities,

more compatible with my strengths and interests. Your attitude had a major positive impact on my life.

Denise Batista, my wonderful wife. You can claim rightful ownership of a big part of this Ph.D. degree. I acknowledge your courage and willingness to embark in this adventure, temporarily putting aside your professional career to invest in mine, confident that it would pay off. The whole new experience, especially at the beginning, required much adaptation, pushing me far beyond my comfort zone. I often doubted my ability and considered giving up. But you were always there, gracefully encouraging me and providing vitally needed emotional support. Later in the program, I would see myself stressed out, worrying about passing the qualifying exam and sometimes seeing my research going backwards rather than progressing. Maintaining mental health through those dark times would have been a major challenge if not, again, for your support and love. On top of that, you surprised me and filled our lives with the most precious gift, our lovely daughter Sophia–who joined us in the second year of our journey, just two weeks after I passed the qualifying exam–and most recently, our baby-to-be Luisa.

All my friends and colleagues from ISyE for much appreciated companionship. Rather than competition, I experienced help and support from my classmates, without which I would not have made it through the course work of the first few semesters of the program. Very special thanks to Alfredo Torrico, Damian Reyes, Luke Marshall, Mina Georgieva, Seyma Guven-Kocak, Timur Tankayev, and Tony Yaacoub, who, countless times, patiently helped me with understanding class materials, doing homework, studying for exams, and preparing for the qualifying exam. Extra thanks to Damian, who first introduced me to Python programming, and most recently, served as a career mentor; to Timur, who always pointed out incredibly simple ways to solve computational issues and find bugs in my codes; to Alfredo for valuable friendship and for all the academic experiences we shared as officemates; to Luke and Seyma for being there during the toughest period of our journey, the qualifying exam; and to Luke again for critical mentoring while I was in the job market.

Eduardo Noronha for challenging me to be nothing less than I was capable of. I was lucky to have met you during my Ph.D. program when I had to find the courage to make what would turn out to be some of the most important decisions of my life. Being involved in one of the projects of your thesis also opened my eyes to the realization that I could actually use optimization to make a substantive impact in the real world. Thank you for letting me be part of that project. I am confident we will continue to make major contributions together.

Yuan Li for your mentorship during the first year. Thank you for your kindness and time invested in helping to make my adaptation process smoother.

My former professors in Brazil, Professor Gustavo da Costa, Professor Clovis Gonzaga, and Professor Antonio Leitao, for your encouragement, recommendation letters, and celebration with me of my achievements along the way.

Andreas Barmann, Braun Gabor, Iman Dayarian, Professor Natashia Boland, and Professor George Nemhauser for the inspirational experience of teaching the Engineering Optimization course in spring 2017. It was such an honor to be part of the team, which helped me significantly increase self-confidence both in teaching and subsequent research. Special thanks to Andreas for all the conversations when you stimulated my thoughts regarding research and challenged me to pursue some of my biggest goals.

Dr. Dawn Strickland, Timur Tankayev, Beste Basciftci, and Professor Santanu Dey for your companionship while teaching the Engineering Optimization course in fall 2018. Because of your enjoyable companionship, grading 200 exams was surprisingly one of the most entertaining parts of teaching the course.

The faculty members with whom I have interacted. From each of you I learned I could use Mathematics to do much more than I would ever have expected. Special thank to the Professors Craig Tovey, Robert Foley, Santanu Dey, Renato Monteiro, David Goldsman, Yajun Mei and He Wang for teaching and inspiring me with your enthusiasm and passion. I am very thankful for having met and learned from such a team of talented people.

My collaborators of side projects for all the experiences exchanged. Special thanks to Joseph Levy, Iman Dayarian, Gustavo Carneiro, Ehsan Khodabandeh and Professor Puneet Dwivedi. My collaboration with each one of you has significantly contributed to enrich my formation. Special thanks also to my coauthors at Emory University, Professor Lars Ruthotto, Professor Ymir Vigfusson and Lauri Mustonen for the great opportunity to learn about a new application and make my modest contribution.

Stacey Doremus and the whole LEAD team. It has been a great pleasure to be part of the team. Learning how to coach and help others grow had a major impact on my own personal development. Special thanks to Stephen Steppe for having coached me in fall 2017, and for having encouraged me to become a coach as well.

My students and coaches for having added so much more meaning to my Ph.D. journey. Thank you for all the great experiences I had while sharing with each of you some of the things learned along the way. Your feedback and words of encouragement convinced me even more that teaching is one thing I was born to do. I will always keep that in mind–as also recommended by Professor George Nemhauser at the day of my thesis defense.

My parents and my sister. I acknowledge your efforts to handle issues on your own, giving me the peace of mind I needed to progress with classes and research. You always encouraged me to pursue my studies and celebrated with me each step completed. This degree is also dedicated to you.

Many other friends, who, with simple attitudes and words, made our life easier and more meaningful over the last few years. Special thanks to Estela, for using one of your precious days off to help us to find a place to live in our first week in the United States, even without knowing us. Special thanks to Marcus, Suzana, Gustavo, Luciana, Guillermo and Maria Paz for your incredible support when we first returned home with our newborn Sophia. Special thanks to Tubal, Gladys, and the Comunitas community for your friendship and support–especially during the first year when you used to take care of Denise while I was studying through the weekend. Special thanks also to Ashish and his wonderful family.

The kindness and love you have shared with us have made your experience even more pleasant and meaningful.

All the people who maintain or have shared their knowledge on Wikipedia and Stack Overflow. I am not ashamed to say that I have reached to these two websites a countless number of times. Wikipedia would always have a clean definition of new terms I needed to learn at the beginning of the program. Stack Overflow would always have a good answer for my questions when I was teaching myself how to code in Python.

CNPq, the sponsor of my scholarship from Brazil (grant number 248941/2013-5). Nothing of this would have been possible without the investment of the Brazilian government. I hope I will be able to reciprocate accordingly. Special thanks to Mr. Standard Silva for effectively helping me address so many concerns I had throughout the program. This work was supported by the NSF CMMI grant numbers 1149400, 1150700, and 1562578; and the NSF grant number 24067A2.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Many classes of mixed integer nonlinear programs (MINLPs) are challenging to solve. A common approach to solve a MINLP is to use a combination of a branch-and-bound algorithm together with convexification and/or cutting-planes. In this thesis, we develop new convexification/cutting-plane techniques for two classes of MINLPs: *mixed integer conic programs* and (non-convex) *quadratically constrained quadratic programs*.

We begin by generalizing a number of important well known results in mixed integer linear programming to the context of mixed integer conic programming. In particular, we introduce a new class of cut generating functions and show that, under some minor technical conditions, these functions, together with integer linear programming-based functions, are sufficient to yield the integer hull of intersections of conic sections in the two-dimensional space.

We then focus on the representability of the convex hull of various sets derived from studying substructures of quadratically constrained quadratic programs. One of our main results shows that the convex hull of a single quadratic constraint intersected with a bounded polyhedron is second-order cone representable. For the bipartite bilinear program, a special case of the quadratically constrained quadratic program, we even design and implement an algorithm to obtain this convex hull. In addition, we introduce a new application of the bipartite bilinear program from civil engineering and report very successful computational results for this instance class.

Finally, we look at the quadratically constrained quadratic program from a rank-1 perspective, i.e., casting the non-convexity of the problem using rank-1 constraints. This approach leads us to identify important substructures from which we derive and convexify several classes of sets. We then apply our results to the well-known pooling problem to obtain successful computational results.

CHAPTER 1

INTRODUCTION AND BACKGROUND

Optimization problems have been part of the scientific world for thousands of years. Especially over the last few decades, optimization algorithms have profoundly changed the way we make practical decisions. In revenue management, they tell us which ads have higher probability of yielding the most profit [32, 132]; in civil engineering, they help us understanding the circumstances under which a given structure, such as a bridge, might collapse [141, 55]; in public health, they help identify strains causing disease [106]; in chemical engineering, they determine the flow of raw material, such as crude oil, from field to refinery [5, 29]; in computer vision, they are used for image recognition [40]; and many other examples could be listed.

While some optimization problems are easy to solve, others are extremely difficult. Finding an efficient algorithm for certain classes of problems poses a major scientific task. Despite the fact that computers and mathematical methods have evolved tremendously in the last few decades, many complex optimization problems arising in our modern society remain unsolved.

Convex Optimization: Perhaps, the most classical (and easy to solve) example of a convex problem is the linear program (LP), which can be formulated as follows:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b \\ & x \in \mathbb{R}^n_+, \end{aligned} \tag{1.1}$$

where A is an $m \times n$ matrix, c and b are vectors of appropriate dimensions and $\mathbb{R}^n_+$ is the set of non-negative vectors in n-dimensional space. LPs are extremely useful for modeling a variety of problems, such as transportation and planning, and can be efficiently solved in practice using the simplex method [21] and interior point methods [96, 129].

However, not every problem can be formulated using only linear expressions. Quadratic terms, for instance, are often needed for modeling basic phenomena emerging from economics [84], biology [106] and engineering [61, 93], just to mention a few. This necessity led to the generalization of the LP known as the conic programming [19], which can be represented as follows:

$$\begin{aligned} \inf \quad & c^\top x \\ \text{s.t.} \quad & Ax - b \in K \\ & x \in \mathbb{R}^n_+, \end{aligned} \tag{1.2}$$

where A is an $m \times n$ matrix, c and b are vectors of appropriate dimensions, and K is a regular cone (i.e., closed, convex, pointed, and full-dimensional) in m-dimensional space. Notice that if $K = \mathbb{R}^n_+$, which is a regular cone, we recover (1.1). Among the most classical examples of conic programs are the semidefinite program (SDP) and the second-order program (SOCP). In the SDP, K is replaced with the cone of symmetric positive semidefinite matrices. In the case of SOCP (which is a special case of the SDP), K is replaced with the second-order cone (Figure 1.1), which is also known as the Lorentz cone:

$$\mathcal{L}^m := \left\{ x \in \mathbb{R}^m \mid \sqrt{x_1^2 + x_2^2 + \cdots + x_{m-1}^2} \leq x_m \right\}.$$

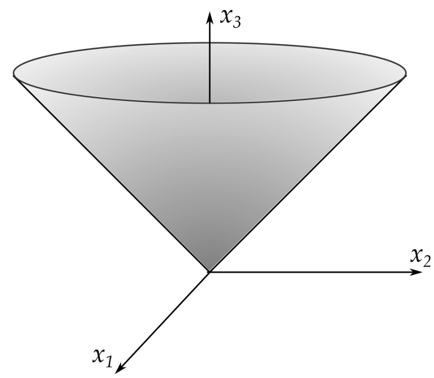

Figure 1.1: The second-order cone in three-dimensional space, also called the ice cream cone.

With fundamental theoretical progress [113, 37, 129] and recent software developments [7, 96, 10, 22], conic programs can be efficiently solved for relatively large instances using interior point methods (often used for solving large scale instances of LPs as well), especially in the case of second-order cone programming [138, 12].

One feature that all convex problems have in common is that every local optimal solution is a global optimal solution (see points A and B in Figure 1.2 for an example of a global and a local solution, respectively).

Non-convex Optimization: Non-convexity in optimization problems are generally due to the presence of discrete variables (binary or integer in most cases) or non-convex expressions (such as the product of two variables), and sometimes both [136]. Different from convex programming, non-convex problems may have multiple local optima (or other stationary points) that are not necessarily a global optimal [112]. This is one of the most troubling aspects of solving non-convex programs.

Figure 1.2 illustrates this fact in a continuous setting. Specifically, the shaded region on the left represents the feasible region of a non-convex problem, and the dashed lines represent a linear objective function that is to be minimized over this region. Therefore, points A and B are both local optimal solutions but only A is a global optimal solution to this problem. The shaded region on the right of Figure 1.2 represents a convex relaxation (not the best one) of the original problem, where C is the minimizer. Observe that C yields a lower bound, zero in this case, on the optimal objective value of the original problem. Solutions A and B yield upper bounds. The difference between the smallest known upper and the largest known lower bound (in the case of a minimization problem) defines the duality gap, which provides a measure of how far we may be from global optimality. For instance, if we only knew solution B, but not A, $2 - 0 = 2$ would be the absolute duality gap in the example of Figure 1.2. The goal is to close the duality gap. Once closed, a provably global optimal solution has been found.

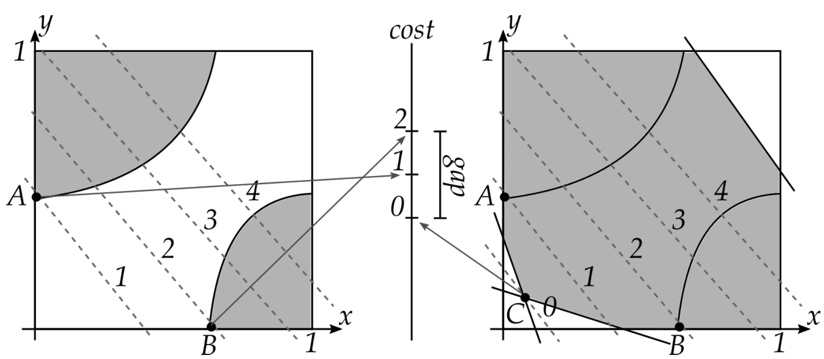

Figure 1.2: Illustration of the feasible region of a non-convex problem along with local and global optimal solutions, a convex relaxation, and the duality gap.

At this point, it should be clear that the ideal convex relaxation would be the smallest convex set that still contains all the feasible solutions to the problem, i.e., the convex hull of the entire set of feasible solutions. However, this is out of reach, except in some special cases. Thus, in reality, convex relaxations by themselves are not enough to solve a non-convex program to global optimality. A very successful approach then, in both discrete and continuous setting, is to use branch-and-bound algorithms [121, 136]. The idea is to partition the feasible region in such a way that: (1) the optimal solution to the previous relaxation is cut off; (2) no feasible solution to the original problem is removed; and (3) the quality of the convex relaxation is improved over each partition. This step is performed recursively for each partition, until the duality gap is closed. It is clear that the mechanism used to derive the convex relaxations plays a major role in the performance of branch-and-bound algorithms.

Convex relaxation can also concern the objective function when it is non-convex. In this case, the goal is to find the tightest convex (or concave in the case of maximization) under-estimate of the original objective function. In fact, historically, convexification of functions has been widely used to derive convex relaxation of the set of feasible solutions as well [44, 1, 68, 131]. For example, the most classical relaxation for the set (which represents the feasible region of a non-convex instance) $S = \{(x, y) \in \mathbb{R}^2 |\ 10xy = 1,\ x, y \in [0, 1]\}$, is derived by rewriting S as $S = \{(x, y, w) \in \mathbb{R}^3 |\ 10w = 1,\ w = xy,\ x, y \in [0, 1]\}$, and then replacing the bilinear term $w = xy$ with its McCormick envelope [4] (function convexification) to obtain:

$$\begin{aligned} S^{Mc} &= \{(x, y, w) \in \mathbb{R}^3 |\ 10w = 1,\ w \geq 0,\ w \geq x + y - 1,\ w \leq x,\ w \leq y,\ x, y \in [0, 1]\} \\ &= \{(x, y, w) \in \mathbb{R}^3 |\ w = 0.1,\ x + y \leq 1.1,\ x \geq 0.1,\ y \geq 0.1,\ x, y \in [0, 1]\}. \end{aligned}$$

However, it is well known that function convexification does not necessarily yield the convex hull of the underlying set. For our example above, we have that (also see Figure 1.3)

$$\mathrm{conv}(S) = \{(x, y, w) \in \mathbb{R}^3 |\ w = 0.1,\ x + y \leq 1.1,\ 10xy \geq 1,\ x, y \in [0, 1]\}.$$

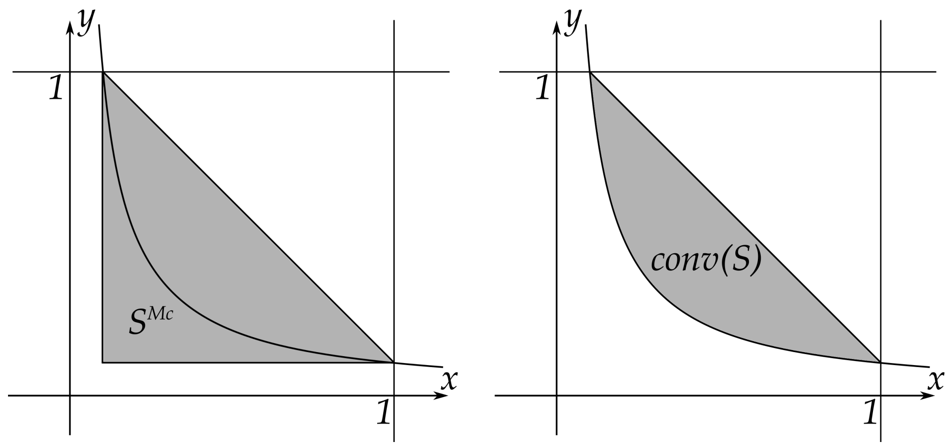

Figure 1.3: Comparison between the McCormick relaxation and the actual convex hull for a simple example. The feasible region is the portion of the curve inside the box.

This motivates our study on convexification of sets rather than on convexification of functions. The computational results of Chapters 3 and 5 show the benefit of our approach.

1.1 Some of our main contributions

Mixed integer linear programs (MILPs), in which all expressions are linear and all the non-convexity comes from the presence of integer variables, have been extensively studied since the 1950's [108, 42]. Both academia and industry have invested tremendous resources in solving MILPs. As a result, solid theory and efficient commercial optimization packages (such as Gurobi, CPLEX and FICO Xpress) are available for solving these problems.

When convex non-linear (conic) expressions are added to a MILP instance, it results in a conic mixed integer program:

$$\begin{aligned} \text{inf} \quad & c^\top x \\ \text{s.t.} \quad & Ax - b \in K \\ & x \in \mathbb{R}^n_+ \\ & x_i \in \mathbb{Z} \; \forall i \in I, \end{aligned} \tag{1.3}$$

where I is the set of integer variables of the problem. This problem is relatively harder to solve, and not as much is known when compared with the MILP. This thesis contributes to the cutting-plane theory for solving conic integer programs. In particular, we show that if the feasible region is bounded, then cut generating functions (see Chapter 2) for integer linear programs can easily be adapted to give the integer hull (i.e., the convex hull of the set of feasible solutions) of the conic integer program. We then introduce a new class of cut generating functions and show that, under some minor technical conditions, these functions, together with integer linear programming-based functions, are sufficient to yield the integer hull of intersections of conic sections in the dimensional space.

In the continuous setting, we focus on the quadratically constrained quadratic program (QCQP), which is an optimization problem of the form:

$$\begin{aligned} \min \quad & x^\top Q^0 x + (a^0)^\top x \\ \text{s.t.} \quad & x^\top Q^k x + (a^k)^\top x \leq b_k \qquad \forall k \in \{1, \ldots, m\} \\ & x \in [0,\ 1]^n, \end{aligned} \tag{1.4}$$

where Q^k for $k \in \{0, \ldots, m\}$ are $n \times n$ matrices, and a^k for $k \in \{0, \ldots, m\}$ and b^k for $k \in \{1, \ldots, m\}$ are vectors of appropriate dimensions. When the matrices Q^i are symmetric positive semi-definite, problem (1.4) reduces to a conic program. We are interested in the general case, i.e., the matrices Q^i's are not assumed to be positive semi-definite and hence the problem is non-convex.

We make several contributions by describing the exact convex hull of various sets that are naturally derived from commonly occurring substructures of the QCQP.

In Chapter 3, we show that the convex hull of the set defined by a single so-called bipartite bilinear equation is second-order cone representable:

Theorem 1.1. *Let* $n_1, n_2 \in \mathbb{Z}_+$, $V_1 \in \{1, \ldots, n_1\}$, $V_2 \in \{1, \ldots, n_2\}$, *and* $E \subseteq V_1 \times V_2$.

Consider the one-constraint set

$$S := \left\{ (x, y, w) \in [0,1]^{n_1+n_2+|E|} \;\middle|\; \begin{array}{l} \sum_{(i,j)\in E} q_{ij} w_{ij} + \sum_{i\in V_1} a_i x_i + \sum_{j \in V_2} b_j y_j + c = 0, \\ w_{ij} = x_i y_j, \; \forall (i,j) \in E \end{array} \right\}.$$

Then $\mathrm{conv}(S)$ *is second-order cone representable.*

The bipartite bilinear program (BBP) is a special case of the QCQP where the variables can be partitioned into two sets such that fixing the variables in any one of the sets results in a linear program. We use our confexification result to propose a convex relaxation for BBP which we show is stronger than the standard SDP and the McCormick relaxations together. We also present an implementable procedure to compute the convex hull of the set S defined in Theorem 1.1. We then implement a branch-and-bound algorithm using this convexification result and test it on instances of the *finite element model updating problem* (see Section 3.2.3), a fundamental problem in structural engineering. Our computational results show that our algorithm significantly outperforms a state-of-the-art commercial global solver in reducing the duality gap for this class of instances.

In Chapter 4, we generalize the result of Theorem 1.1. Specifically, by establishing a connection with the theory of ruled surfaces [57], we show the following result:

Theorem 1.2. *Let*

$$S := \{x \in \mathbb{R}^n \mid \; x^\top Q x + \alpha^\top x = g, \; x \in P\},$$

where $Q \in \mathbb{R}^{n\times n}$ *is a symmetric matrix,* $\alpha \in \mathbb{R}^n$*,* $g \in \mathbb{R}$ *and* $P := \{x \,|\, Ax \leq b\}$ *is a polytope. Then* $\mathrm{conv}(S)$ *is second-order cone representable.*

Despite the generality of the result, the proof we provide for Theorem 1.2 is constructive and fairly simple.

In Chapter 5, we look at the QCQP from a different perspective. Specifically, we use a rank-1 constraint to rewrite the QCQP defined in (1.4) as follows:

$$
\begin{aligned}
\min \quad & \langle Q^0, X\rangle + (a^0)^\top x && (1.5)\\
\text{s.t.} \quad & \langle Q^k, X\rangle + (a^k)^\top x \le b_k \qquad \forall k \in \{1,\ldots,m\} && (1.6)\\
& \operatorname{rank}\left(\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix}\right) = 1 && (1.7)\\
& x \in [0,\ 1]^n, && (1.8)
\end{aligned}
$$

where $\langle U, V\rangle := \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} U_{ij}V_{ij}$, which is the same as the trace inner product in the case of symmetric matrices. Observe that all the non-convexity of the problem is now captured by the rank-1 condition. This motivates our study of sets defined by a rank-1 constraint together with some linear side constraints:

$$
\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]_{k=1}^m) := \left\{W \in \mathbb{R}_+^{n_1\times n_2} \,\middle|\, \langle A^k, W\rangle \le b_k, \forall k \in \{1,\ldots,m\},\ \operatorname{rank}(W) \le 1\right\},
$$

where we can recover (1.6)–(1.8) by replacing W with $\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix}$ and choosing A^ks appropriately.

In this case, we show that the convex hull of $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]_{k=1}^m)$ is second-order cone representable for many choices of the linear side constraints. We also show that some of these choices come up naturally in the application of the pooling problem [67] and we use them to derive new convex relaxations for the generalized pooling problem. Finally, we run a comprehensive set of computational experiments and show that our convexification results together with discretization significantly help in improving dual bounds for the generalized pooling problem.

CHAPTER 2
SOME CUT-GENERATING FUNCTIONS FOR SECOND-ORDER CONIC SETS

The work presented in this chapter has already been published [123].

2.1 Introduction: Subadditive dual of conic integer programs

A natural generalization of linear integer programming is *conic integer programming*. Given a *regular* cone $K \subseteq \mathbb{R}^n$, that is a cone that is pointed, closed, convex, and full dimensional, we can define a conic integer program as:

$$\begin{aligned} \inf \quad & c^\top x \\ \text{s.t.} \quad & Ax - b \in K \\ & x \in \mathbb{Z}^n_+, \end{aligned} \tag{2.1}$$

where $A \in \mathbb{R}^{m\times n}, c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. As is standard, we will henceforth write the constraint $Ax - b \in K$ as $Ax \succeq_K b$, where we use the notation that $u \succeq_K v$ if and only if $u - v \in K$. In the case where K is the non-negative orthant, that is $K = \mathbb{R}^m_+$, the conic integer program is a standard linear integer program.

A natural way to generate cuts for conic integer programs is via the notion of cut-generating functions [43]. Consider a function $f : \mathbb{R}^m \to \mathbb{R}$ that satisfies the following:

1. f is *subadditive*, that is $f(u) + f(v) \geq f(u + v)$ for all $u, v \in \mathbb{R}^m$,

2. f is *non-decreasing with respect to* K, that is $f(u) \geq f(v)$ whenever $u \succeq_K v$,

3. $f(0) = 0$.

Then it is straightforward to see that the inequality

$$\sum_{j=1}^{n} f(A^j)x_j \geq f(b), \tag{2.2}$$

is valid for the conic integer program (2.1), where A^j is the j-th column of A. We denote the set of functions satisfying (1.), (2.) and (3.) above as $\mathcal{F}_K$.

In the paper [105], it was shown that, assuming a technical '*discrete Slater*' condition holds, the closure of the convex hull of the set of integer feasible solutions to (2.1) is described by inequalities of the form (2.2) obtained from $\mathcal{F}_K$. This result from [105] generalizes result on subadditive duality of linear integer programs [74, 75, 76, 139], that is inequalities (2.2) give the convex hull of (2.1) when $K = \mathbb{R}^m_+$ and the constraint matrix A is rational. Also see [80, 78] for related models and results.

In the case where $K = \mathbb{R}^m_+$ and assuming A is rational, a lot more is known about the subset of functions from $\mathcal{F}_{\mathbb{R}^m_+}$ that are sufficient to describe the convex hull of integer solutions (also called as the integer hull). For example, these functions have a constructive characterization using the Chvátal-Gomory procedure [24], it is sufficient to consider functions that are applied to every 2^n subset of constraints at a time (see [124], Theorem 16.5), or for a fixed A there is a finite list of functions independent of b that describes the integer hull [139].

The main goal of this chapter is to similarly better understand structural properties of subsets of functions from $\mathcal{F}_K$ that are sufficient to produce the integer hull of the underlying conic representable set $\{x \in \mathbb{R}^n \,|\, Ax \succeq_K b\}$.

2.2 Main results

We will refer to the *dual cone* of a cone K as K^* which we remind the reader is the set $K^* := \{y \in \mathbb{R}^m \,|\, y^\top x \geq 0 \; \forall x \in K\}$. Given a positive integer m, we denote the set $\{1, \ldots, m\}$ by $[m]$. And given a subset X of $\mathbb{R}^n$ we denote its integer hull by X^I.

2.2.1 Bounded sets

Given a regular cone K we call as *linear composition* the set of functions f obtained as follows: Let the vectors $w^1, w^2, \ldots, w^p \in K^*$ and the function $f : \mathbb{R}^m \to \mathbb{R}$ be given by

$$f(v) = g((w^1)^\top v, (w^2)^\top v, \cdots, (w^p)^\top v), \tag{2.3}$$

where $g \in \mathcal{F}_{\mathbb{R}^p_+}$ satisfies $g(u) = -g(-u)$ for all $u \in \mathbb{R}^p$. It is straightforward to see that linear composition functions belong to $\mathcal{F}_K$ and also satisfy $f(v) = -f(-v)$ for all $v \in \mathbb{R}^m$, which implies that f generates valid inequalities of the form (2.2) even when the variables are not required to be non-negative. Our first result describes a class of conic sets for which linear composition functions are sufficient to produce the convex hull.

Theorem 2.1. *Let $K \subseteq \mathbb{R}^m$ be a regular cone. Consider the conic set $T = \{x \in \mathbb{R}^n \mid Ax \succeq_K b\}$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Assume T has nonempty interior. Let $\pi^\top x \geq \pi_0$ be a valid inequality for T^I where $\pi \in \mathbb{Z}^n$ is non-zero. Assume $B := \{x \in T \mid \pi^\top x \leq \pi_0\}$ is nonempty and bounded. Then, for some natural number $p \leq 2^n$, there exist vectors $y^1, y^2, \ldots, y^p \in K^*$ such that $\pi^\top x \geq \pi_0$ is a valid inequality for the integer hull of the polyhedron $Q = \{x \in \mathbb{R}^n \mid (y^i)^\top Ax \geq (y^i)^\top b,\ i \in [p]\}$, where $(y^i)^\top A$ is rational for all $i \in [p]$.*

We highlight here that particular care was taken in Theorem 2.1 to ensure that the outer approximating polyhedron has rational constraints.

Since a valid inequality for Q^I can be obtained using a subadditive function $g \in \mathcal{F}_{\mathbb{R}^p_+}$ that satisfies $g(u) = -g(-u)$ for all $u \in \mathbb{R}^p$ [109] (note that the constraints matrix defining Q is rational), Theorem 2.1 implies that if a cut separates a bounded set from T, then it can be obtained using exactly one function (2.3) with $p \leq 2^n$. Geometrically, Theorem 2.1 can be interpreted as the fact that if the set of points separated is bounded, then the cut can be obtained using a rational polyhedral outer approximation.

We obtain the following corollary immediately: If the set $\{x \in \mathbb{R}^n \mid Ax \succeq_K b\}$ is compact and has non-empty interior, then it is sufficient to restrict attention to linear composition functions to obtain the convex hull. A proof of Theorem 2.1 is presented in Section 2.3.

2.2.2 New family of cut-generating functions

In the previous section we stated that any valid inequality for the integer hull of a bounded conic set can be obtained using linear composition functions. So what happens when the underlying set is not bounded? Consider the simple unbounded set $T' = \{(x_1, x_2) \in \mathbb{R}^2_+ \mid x_1 x_2 \geq 1\}$, which is one branch of a hyperbola[1]. This set is conic representable, that is $T' = \{x \in \mathbb{R}^2_+ \mid Ax \succeq_K b\}$, where K is the second-order cone $\mathcal{L}^3$ and

$$A = \begin{bmatrix} 0 & 0 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix}. \tag{2.4}$$

(We use the notation $\mathcal{L}^m := \left\{x \in \mathbb{R}^m \mid \sqrt{x_1^2 + x_2^2 + \cdots + x_{m-1}^2} \leq x_m\right\}$ to represent the second-order cone in $\mathbb{R}^m$.) The integer hull of T' is given by the following two inequalities:

$$x_1 \geq 1, \; x_2 \geq 1. \tag{2.5}$$

It is straightforward to verify that the inequalities (2.5) are *not valid for any polyhedral outer approximation of* T'. Indeed any polyhedral outer approximation of T' contains integer points not belonging to T' (see Proposition 3). Therefore, applying the cut-generating recipe (2.3) a finite number of times (that is considering integer hulls of a finite number of polyhedral outer approximations of T') does not yield $x_1 \geq 1$. However, we note here that we can use linear composition (2.3) to obtain a cut of the form $x_1 + x_2/k \geq 1$ where

[1] In this chapter, we refer to the curve, as well as the convex region delimited by this curve, as the branch of a hyperbola. Same for parabolas and ellipses.

$k \in \mathbb{Z}_+$ and $k \geq 1$. Clearly

$$\bigcap_{k \in \mathbb{Z}_+, k \geq 1} \left\{x \in \mathbb{R}^2 \,|\, x_1 + x_2/k \geq 1\right\} = \{x \in \mathbb{R}^2 \,|\, x_1 \geq 1\}.$$

However, it would be much nicer if we could *directly obtain* $x_1 \geq 1$ without resorting to obtaining it as an implication of an infinite sequence of cuts.

Many papers [63, 64, 47, 50, 54, 117, 116, 16, 85] have explored various families of subadditive functions for linear integer programs. Our second result, in the same spirit, is a parametrized family of functions that belongs to $\mathcal{F}_K$, where K is the second-order cone $\mathcal{L}^m$. The formal result is as follows:

Theorem 2.2. *Let $j \in [m-1]$. Define $\Gamma_j := \{\gamma \in \mathbb{R}^m \,|\, \gamma_m \geq \sum_{i=1}^{m-1} |\gamma_i|,\ \gamma_m > |\gamma_j|\}$. Suppose $\gamma \in \Gamma_j \cup \operatorname{int}(\mathcal{L}^m)$. Consider the real-valued function $f_\gamma : \mathbb{R}^m \to \mathbb{R}$ defined as:*

$$f_\gamma(v) = \begin{cases} \gamma^\top v + 1 & \text{if } v_j \neq 0 \text{ and } \gamma^\top v \in \mathbb{Z}, \\ \lceil \gamma^\top v \rceil & \text{otherwise}. \end{cases} \tag{2.6}$$

Then, $f_\gamma \in \mathcal{F}_{\mathcal{L}^m}$.

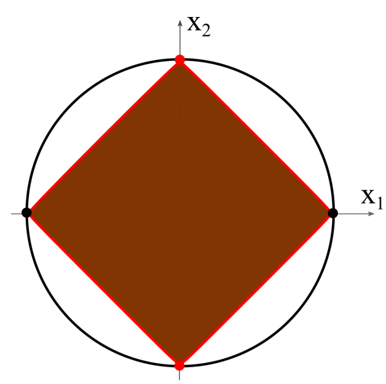

Figure 2.1: Slice at $x_3 = 1$ of the second-order cone $\mathcal{L}^3$ and Γ_1.

To see an example of use of f_γ, consider $j = 1$ and $\gamma = (0, 0.5, 0.5)$. Then applying the resulting function f_γ to the columns of (2.4) we obtain the inequality $x_1 \geq 1$.

Note that the validity of the first inequality in (2.5) can be explained via the disjunction $x_1 \leq 0 \vee x_1 \geq 1$. Therefore, some of the cuts generated using (2.6) can be viewed as split disjunctive cuts. Significant research has gone into describing split disjunctive cuts (newer implied conic constraints) for conic sections [46, 17, 102, 103, 79, 140, 34]. However, to the best of our knowledge, there is no family of subadditive functions in $\mathcal{F}_{\mathcal{L}^m}$ which have been described in closed form previously.

It is instructive to compare cuts obtained using (2.6) with two well-known approaches for generating cuts for the integer hull of second-order conic sets [39, 11]. Note that the CG cuts described in [39] are a special case[2] of cuts generated via linear composition (2.3). Therefore as discussed above, the CG cuts described in [39] cannot generate (2.5) directly. The conic MIR procedure described in [11] begins with first generating an extended formulation which applied to T' would be of the form:

$$\begin{aligned} t_0 &\leq x_1 + x_2 \\ t_1 &\geq 2 \\ t_2 &\geq |x_1 - x_2| \\ t_0 &\geq \|t\|_2 \\ & \quad x_1, x_2 \in \mathbb{Z}_+, t \in \mathbb{R}^3_+. \end{aligned}$$

Then, cuts for the set $\{(x, t_2) \in \mathbb{Z}^2_+ \times \mathbb{R} \,|\, t_2 \geq |x_1 - x_2|\}$ are considered. However, this set is integral in this case and therefore no cuts are obtained. Thus, the conic MIR procedure does not generate the inequalities (2.5).

Remark 1. *The function f_γ defined in (2.6) is piecewise linear, and it is therefore tempting to think it may also belong to $\mathcal{F}_{\mathcal{R}^m_+}$. However it is straightforward to check that f_γ is not necessarily non-decreasing with respect to $\mathbb{R}^3_+$. Let $j = 1$ and $\gamma = (0, \rho, \rho)$ where ρ is a*

[2] More precisely, in [39] the variables are assumed to be non-negative, in which case we can drop the requirement of g satisfying $g(u) = -g(-u)$ in the definition of linear composition.

positive scalar. Then

$$f_\gamma(v_1, v_2, v_3) = \begin{cases} \rho(v_2+v_3)+1 & \text{if } v_1 \neq 0 \text{ and } \rho(v_2+v_3) \in \mathbb{Z}, \\ \lceil \rho(v_2+v_3) \rceil & \text{otherwise.} \end{cases}$$

Consider the vectors $u = (0, 0, 1/\rho)$ *and* $v = (-1, 0, 1/\rho)$. *Then* $u \geq_{\mathbb{R}^3_+} v$, *whereas* $f_\gamma(u) = 1 < 2 = f_\gamma(v)$.

A proof of Theorem 2.2 is presented in Section 2.4.

2.2.3 Cuts for integer conic sets in $\mathbb{R}^2$

As mentioned earlier, the family of functions (2.6) yields the inequalities (2.5). Indeed, we are able to verify a more general result in $\mathbb{R}^2$. To explain this result, we will need the following results:

Lemma 1. *Let* G *be one branch of a hyperbola in* $\mathbb{R}^2$. *Then* G *can be represented as* $G = \{x \in \mathbb{R}^2 \mid Ax \succeq_{\mathcal{L}^3} b\}$, *where* $A \in \mathbb{R}^{3\times 2}$ *is such that* $A_{11}, A_{12} = 0$. *Moreover, the asymptotes of* G *have equations*

$$(A_{21} + A_{31})x_1 + (A_{22} + A_{32})x_2 = b_3 + b_2 \tag{2.7}$$

$$(-A_{21} + A_{31})x_1 + (-A_{22} + A_{32})x_2 = b_3 - b_2. \tag{2.8}$$

In order to generate cuts for G in Lemma 1 using functions (2.6) we first require the variables to be non-negative. Therefore, let us write G as

$$A^1x_1^+ - A^1x_1^- + A^2x_1^+ - A^2x_1^- \succeq_{\mathcal{L}^3} b \tag{2.9}$$

$$x_1^+, x_1^-, x_2^+, x_2^- \geq 0 \tag{2.10}$$

$$x_j = x_j^+ - x_j^- \; j \in \{1, 2\}. \tag{2.11}$$

Assuming that the asymptotes of G are rational, we may assume that the coefficients in (2.7) and (2.8) are integers and then let $\tau = \gcd(A_{21} + A_{31}, A_{22} + A_{32})$. Let $j = 1$ and $\gamma = (0, 1/\tau, 1/\tau)$. Then we apply the function f_γ to obtain the following cut for (2.9), (2.10):

$$\frac{(A_{21} + A_{31})}{\tau} x_1^+ - \frac{(A_{21} + A_{31})}{\tau} x_1^- + \frac{(A_{22} + A_{32})}{\tau} x_2^+ - \frac{(A_{22} + A_{32})}{\tau} x_2^- \geq f_\gamma(b). \quad (2.12)$$

Now, using (2.11) and observing that the coefficient of x_j^+ is the negative of the coefficient of x_j^- in (2.12), $j = 1, 2$, we can project the inequality (2.12) to the space of the original x variables. The resulting cut is parallel to the asymptote (2.7). We can do a similar calculation to obtain a cut parallel to the other asymptote (2.8). We state all this concisely in the next proposition.

Proposition 1. *Let $G = \{x \in \mathbb{R}^2 \mid Ax \succeq_{\mathcal{L}_+^3} b\}$ be one branch of a hyperbola with rational asymptotes, where $A \in \mathbb{R}^{3\times 2}$ and $A_{11}, A_{12} = 0$. Then the following inequalities are valid for G^I:*

$$(u^j)^\top A^1 x_1 + (u^j)^\top A^2 x_2 \geq \tau^j f_{\gamma^j}(b), \quad (2.13)$$

where $u^1 = (0, 1, 1)$, $u^2 = (0, -1, 1)$, $\tau^j = \gcd((u^j)^\top A^1, (u^j)^\top A^2)$ and $\gamma^j := u^j/\tau^j$, $j = 1, 2$.

We are now ready to state the main result of this section.

Theorem 2.3. *Let $W = \bigcap_{i\in[m]} W^i$, where $W^i = \{x \in \mathbb{R}^2 \mid A^i x \succeq_{\mathcal{L}^{m_i}} b^i\}$, $A^i \in \mathbb{R}^{m_i\times 2}$, $b^i \in \mathbb{R}^{m_i}$ and $\mathcal{L}^{m_i}$ is the second-order cone in $\mathbb{R}^{m_i}$. Assume W has nonempty interior and each constraint $A^i x \succeq_{\mathcal{L}^{m_i}} b^i$ in the description of W is either a half-space or a single conic section, such as a parabola, an ellipse, or one branch of a hyperbola. Also assume that if W^i is a hyperbola, then it is non-degenerate and it is written as in Lemma 1, that is $A^i \in \mathbb{R}^{3\times 2}$ and $A^i_{11}, A^i_{12} = 0$. Finally, we assume that each W^i is non-redundant, that is,* *for all $j \in [m]$, W is strictly contained in $\bigcap_{i\in[m],i\neq j} W^i$. Then the following statements hold:*

1. *If $W \cap \mathbb{Z}^2 = \emptyset$, then this fact can be certified with the application of at most two inequalities generated from (2.3) or (2.13);*

2. *Assume* $\text{interior}(W) \cap \mathbb{Z}^2 \neq \emptyset$*. If $\pi^\top x \geq \pi_0$ defines a face of W^I where $\pi \in \mathbb{Z}^2$ is non-zero, then this inequality can be obtained with application of exactly one function (2.3) or it is one of the inequalities (2.13).*

Proof of Lemmma 1 and Theorem 2.3 are presented in Section 2.5.

2.3 Cutting-planes separating bounded set of points

In this section, we prove Theorem 2.1. We begin by stating three well-known lemmas.

Lemma 2. *Let $K \in \mathbb{R}^n$ be a closed cone and let K^* denote its dual. Then* $\text{int}\,(K^*) = \{y \in \mathbb{R}^n \mid y^\top x > 0 \; \forall x \in K \setminus \{0\}\}$.

Hereafter, we will denote the recession cone of a set C by rec.cone(C) and the dual of rec.cone(C) by rec.cone$^*(C)$.

Lemma 3. *Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Then the following statements hold:*

(i) for every $c \in$ int (rec.cone$^*(C)$) *the problem* $\inf\{c^\top x \mid x \in C\}$ *is bounded.*

(ii) for every $c \notin$ rec.cone$^*(C)$ *the problem* $\inf\{c^\top x \mid x \in C\}$ *is unbounded.*

Lemma 4 (Conic strong duality [19])**.** *Let $K \subseteq \mathbb{R}^m$ be a regular cone. Consider the conic set $T = \{x \in \mathbb{R}^n \mid Ax \succeq_K b\}$, where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Assume* $\text{int}\, T \neq \emptyset$*. If $c \in \mathbb{R}^n$ is such that* $\inf\{c^\top x \mid x \in T\}$ *is bounded, then there exists $y \in K^*$ such that $y^\top A = c^\top$ and* $y^\top b = \inf\{c^\top x \mid x \in T\}$.

The next lemma states that under some conditions it is possible to separate a point from a set using a rational separating hyperplane.

Lemma 5. *Let $C \subseteq \mathbb{R}^n$ be a closed convex set. Assume* $\operatorname{int}(\text{rec.cone}^*(C)) \neq \emptyset$. *Let $z \notin C$. Then there exist $\pi \in \mathbb{Q}^n$, $\pi \neq 0$, and $\pi_0 \in \mathbb{R}$ such that $\pi^\top z < \pi_0 \leq \pi^\top x$ for all $x \in C$.*

Proof. The standard separation theorem ensures that there exist $w \in \mathbb{R}^n$, $w \neq 0$, and $w_0 \in \mathbb{R}$ such that $w^\top z < w_0 \leq w^\top x$ for all $x \in C$. As $\operatorname{int}(\text{rec.cone}^*(C)) \neq \emptyset$ there exist $w^1, w^2, \ldots, w^{n+1} \in \operatorname{int}(\text{rec.cone}^*(C))$ affinely independent. For every $i \in [n+1]$ let $w_0^i = \inf\{(w^i)^\top x \mid x \in C\}$. In view of Lemma 3 we have that w_0^i is finite for all $i \in [n+1]$. Since $w_0 - w^\top z > 0$ and z is fixed, we can chose $\varepsilon_i > 0$, $i \in [n+1]$, such that

$$\left| \sum_{i=1}^{n+1} \varepsilon_i (w^i)^\top z - \sum_{i=1}^{n+1} \varepsilon_i w_0^i \right| < w_0 - w^\top z. \tag{2.14}$$

Moreover, since $w^1, w^2, \ldots, w^{n+1}$ are affinity independent, the cone generated by these vectors is full dimensional. Thus, the scalars $\varepsilon_i > 0$, $i \in [n+1]$, can be chosen such that $\pi := w + \sum_{i=1}^{n+1} \varepsilon_i w^i \in \mathbb{Q}^n$. Now observe that

$$\begin{aligned}\pi^\top z < w_0 + \sum_{i=1}^{n+1} \varepsilon_i w_0^i &\leq \inf\{w^\top x \mid x \in C\} + \sum_{i=1}^{n+1} \inf\{(\varepsilon_i w^i)^\top x \mid x \in C\} \\ &\leq \inf\left\{ \left(w^\top + \sum_{i=1}^{n+1} \varepsilon_i w^i \right)^\top x \mid x \in C \right\} \leq \pi^\top x \ \ \forall x \in C,\end{aligned}$$

where the first strict inequality follows from (2.14). Therefore, $\pi^\top z < \pi_0 \leq \pi^\top x$ for all $x \in C$, where $\pi_0 := w_0 + \sum_{i=1}^{n+1} \varepsilon_i w_0^i$. $\square$

The next result will imply Theorem 2.1.

Proposition 2. *Let T be the set as in the statement of Lemma 4. Consider the set $B := \{x \in T \mid \pi^\top x \leq \pi_0\}$, where $\pi \in \mathbb{Z}^n$ is non-zero. Then B is bounded if and only if $\pi \in$* interior(rec.cone$^*(T)$)*, in which case for some natural number p', there exist vectors $y^1, y^2 \ldots, y^{p'} \in K^*$ such that the polyhedron*

$$P = \{x \in \mathbb{R}^n \mid \pi^\top x \leq \pi_0,\ (y^i)^\top A x \geq (y^i)^\top b,\ i \in [p']\}$$

contains B and $P^I = B^I$, where $(y^i)^\top A$ is rational for all $i \in [p']$.

Proof. Assume B is bounded. We claim that $d^\top \pi > 0,$ for all $d \in$ rec.cone$(T) \setminus \{0\}$. Indeed, if $d \in$ rec.cone(T) is such that $d^\top \pi \leq 0$, then $d \in$ rec.cone(B), which implies that $d = 0$ since B is bounded. Now, in view of Lemma 2, the claim implies that $\pi \in$ int (rec.cone$^*(T)$).

Assume $\pi \in$ int (rec.cone$^*(T)$). As $\pi \in \mathbb{Z}^n$, let $\{v^1, v^2, \dots, v^{n-1}\} \subseteq \mathbb{Q}^n$ be an orthogonal basis of the linear subspace orthogonal to π. Since $\pi \in$ int (rec.cone$^*(T)$), there exists a positive constant ε such that $w^i := \pi + \varepsilon v^i$ and $w^{i+n-1} := \pi - \varepsilon v^i$ belong to int (rec.cone$^*(T)$) for all $i \in [n-1]$. As we may assume that ε is rational, we obtain that w^i is rational for all $i \in [2n-2]$. It follows from Lemma 3 and Lemma 4 that for all $i \in [2n-2]$ there exists $y^i \in K^*$ such that $(y^i)^\top Ax \geq (y^i)^\top b$ is a valid inequality for T, where $(y^i)^\top A = w^i \in \mathbb{Q}^n$. Since $\pi \in$ int (rec.cone$^*(T)$), Lemma 3 and Lemma 4 also imply that there exists $y^{2n-1} \in K^*$ such that $(y^{2n-1})^\top Ax \geq (y^{2n-1})^\top b$ is a valid inequality for T, where $(y^{2n-1})^\top A = \pi^\top \in \mathbb{Q}^n$. Now, let $P^1 = \{x \in \mathbb{R}^n \mid \pi^\top x \leq \pi_0,\ (y^i)^\top Ax \geq (y^i)^\top b,\ i \in [2n-1]\}$. By our choice of w^i and using the fact that $(y^{2n-1})^\top b \leq \pi^\top x \leq \pi_0$ for all $x \in P^1$ (if $\pi_0 \leq (y^{2n-1})^\top b$, then $P^1 = \emptyset$), it is easy to verify that P^1 is bounded. Since P^1 contains B, we obtain that B is also bounded.

If $(P^1)^I = B^I$, then we are done by setting P to P^1, in which case $p' = 2n-1$. Otherwise, as P^1 is bounded, there is only a finite number of integer points $z \in P^1 \setminus B$. For each one of these points z, we construct a rational valid inequality $w_0 \leq w^\top x$ for T that is guaranteed by Lemma 5 that separates z from B, that is $w^\top z < w_0$. It remains to show that this inequality can be obtained 'via dual multipliers': This is straightforward by again examining the conic program $\inf\{w^\top x \mid x \in T\}$ and applying Lemma 4. □

Proof. of Theorem 2.1 Let $\pi^\top x \geq \pi_0$ be a valid inequality for T^I, where $\pi \in \mathbb{Z}^n$ is non-zero. Suppose $B = \{x \in T \mid \pi^\top x \leq \pi_0\}$ is nonempty and bounded. Then, by Proposition 2, using dual multipliers $y^0, y^1, \dots, y^{p'} \in K^*$, and letting $P = \{x \in \mathbb{R}^n \mid \pi^\top x \leq \pi_0,\ (y^i)^\top Ax \geq (y^i)^\top b,\ i \in [p']\}$, we have that (i) $P \supseteq B$ and (ii) $P \cap \mathbb{Z}^n = B \cap \mathbb{Z}^n$. Note that interior$(B) \cap \mathbb{Z}^n = \emptyset$ and the only integer points in B are those that satisfy $\pi^\top x = \pi_0$.

Now using an argument similar to Corollary 16.5a [124], there is a subset of 2^n inequalities defining P together with $\pi^T x < \pi_0$ such that the resulting set contains no integer points. WLOG $\{x \in \mathbb{R}^n \mid \pi^\top x \leq \pi_0,\ (y^i)^\top Ax \geq (y^i)^\top b,\ i \in [p]\}$ is lattice-free, where $p \leq 2^n$, i.e., $\pi^\top x \geq \pi_0$ is a valid inequality for the integer hull of $Q = \{x \in \mathbb{R}^n \mid (y^i)^\top Ax \geq (y^i)^\top b,\ i \in [p]\}$ where $(y^i)^\top A \in \mathbb{Q}^n$ for $i \in [p]$. $\square$

Remark 2. *If $T \cap \mathbb{Z}^n \neq \emptyset$, then using the same argument as in the proof of Corollary 16.6 [124] (also see [73]), the bound of 2^n in Theorem 2.1 can be improved to $2^n - 1$.*

The next proposition illustrates that if the set B in the statement of Theorem 2.1 is not bounded, then the result may not hold.

Proposition 3. *Let $T' := \{(x \in \mathbb{R}^2_+ \mid x_1 x_2 \geq 1\}$. Every polyhedral outer approximation of T' contains points of the form $(0, k)$ (and similarly points of form $(k, 0)$) for k sufficiently large natural number.*

Proof. Suppose $\{x \in \mathbb{R}^2 \mid \alpha_1^i x_1 + \alpha_2^i x_2 \geq \beta_i,\ i \in [q]\}$, is a polyhedral outer approximation of T' where q is some natural number. Since the recession cone of this polyhedron contains the recession cone of T', that is $\mathbb{R}^2_+$, we have that $\alpha_1^i, \alpha_2^i \geq 0$.

We will prove that there exist points of the form $(0, k)$ belonging to this outer approximation by showing that for all $i \in [q]$ there exists a k_i such that $(\alpha^i)^\top (0, t) \geq \beta_i$ for all $t \in [k_i, \infty) \cap \mathbb{Z}$. If $\alpha_2^i = 0$, then $\beta_i \leq 0$ (since $\alpha_1^i / k + \alpha_2^i k \geq \beta_i$ for all $k \in \mathbb{R}_+$). Therefore $k_i = 0$. If $\alpha_2^i > 0$, then $k_i = \beta_i / \alpha_2^i$. $\square$

2.4 A family of cut-generating functions in $\mathcal{F}_{\mathcal{L}^m}$ and its properties

In this section, we show that f_γ defined in (2.6) belongs to $\mathcal{F}_K$. Clearly f_γ satisfies property (3.) in the definition of $\mathcal{F}_K$, that is $f_\gamma(0) = 0$. In Proposition 4 and 5 we prove that f_γ also satisfies properties (1.) and (2.).

Proposition 4. *The function f_γ defined in (2.6) is subadditive.*

Proof. Let $u, v \in \mathbb{R}^m$. If at least one of these vectors fits in the first clause of (2.6), then we have

$$f_\gamma(u+v) \leq \lceil \gamma^\top (u+v) \rceil + 1 \leq \lceil \gamma^\top u \rceil + \lceil \gamma^\top v \rceil + 1 \leq f_\gamma(u) + f_\gamma(v).$$

Now, suppose that neither u nor v satisfies the first clause. If $u+v$ does not fit in the first clause, then we are done because $\lceil \cdot \rceil$ is a subadditive function. Assume $u+v$ satisfies the first clause, that is

$$u_j + v_j \neq 0, \;\; \gamma^\top (u+v) = \gamma^\top u + \gamma^\top v \in \mathbb{Z}. \tag{2.15}$$

In this case, u_j and v_j cannot be simultaneously zero, say $u_j \neq 0$. Then

$$\gamma^\top u \notin \mathbb{Z}, \tag{2.16}$$

because u does not satisfies the first clause. It follows from (2.15) and (2.16) that

$$\gamma^\top v \notin \mathbb{Z}. \tag{2.17}$$

Finally, (2.15), (2.16), (2.17) together imply

$$f_\gamma(u) + f_\gamma(v) = \lceil \gamma^\top u \rceil + \lceil \gamma^\top v \rceil = \gamma^\top u + \gamma^\top v + 1 = f_\gamma(u+v),$$

where the second inequality follows from the fact that $\gamma^\top u + \gamma^\top v \in \mathbb{Z}$. $\square$

Lemma 6. *Let $w \in \mathcal{L}^m$ and $j \in [m-1]$. Let Γ_j be the set as in the statement of Theorem 2.2. If $\gamma \in \mathcal{L}^m$, then $\gamma^\top w \geq 0$. If, in addition, $\gamma \in \Gamma_j \cup \operatorname{int}(\mathcal{L}^m)$ and $w_j \neq 0$, then $\gamma^\top w > 0$.*

Proof. We have that $\gamma \in \mathcal{L}^m$. Therefore, since $w \succeq_{\mathcal{L}^m} 0$ and $\mathcal{L}^m$ is a self-dual cone, we conclude that $\gamma^\top w \geq 0$. Now, assume $w_j \neq 0$. If either γ or w is in the interior of $\mathcal{L}^m$, then it follows directly from Lemma 2 that $\gamma^\top w > 0$. Assume $\gamma, w \notin \operatorname{int}(\mathcal{L}^m)$. Then

$$w_m = \sqrt{w_1^2 + w_2^2 + \cdots + w_{m-1}^2} \tag{2.18}$$

$$\gamma_m = \sqrt{\gamma_1^2 + \gamma_2^2 + \cdots + \gamma_{m-1}^2}. \tag{2.19}$$

Two observations follows: (i) as $w_j \neq 0$, equation (2.18) implies that for all $i \in [m-1]$ such that $i \neq j$ we have $w_m > |w_i|$; (ii) since $\gamma_m > |\gamma_j|$, equation (2.19) implies that $\gamma_i \neq 0$ for some $i \in [m-1]$ such that $i \neq j$. Now, for all $i \in [m-1]$ such that $\gamma_i \geq 0$, we multiply $w_m > -w_i$ by γ_i and, for all $i \in [m-1]$ such that $\gamma_i < 0$, we multiply $w_m > w_i$ by $-\gamma_i$. In view of observations (i) and (ii), at least one of the resulting inequalities remains strict. Then adding them all we obtain

$$\begin{aligned} \sum_{i \in [m-1]:\, \gamma_i \geq 0} \gamma_i w_m + \sum_{i \in [m-1]:\, \gamma_i < 0} -\gamma_i w_m &> \sum_{i \in [m-1]:\, \gamma_i \geq 0} \gamma_i(-w_i) + \sum_{i \in [m-1]:\, \gamma_i < 0} (-\gamma_i) w_i \\ \Rightarrow \sum_{i \in [m-1]} |\gamma_i| w_m &> -\sum_{i \in [m-1]} \gamma_i w_i \\ \Rightarrow \gamma_m w_m &> -\sum_{i \in [m-1]} \gamma_i w_i, \end{aligned}$$

where the last implication follows from the fact that $\gamma_m \geq \sum_{i=1}^{m-1} |\gamma_i|$ and $w_m \geq 0$. The result follows from this last inequality. $\square$

Proposition 5. *The function f_γ defined in (2.6) is non-decreasing with respect to $\mathcal{L}^m$.*

Proof. Let $u, v \in \mathbb{R}^m$. Suppose $u \succeq_{\mathcal{L}^m} v$. By applying Lemma 6 to $w = u - v$ we conclude that

$$\gamma^\top u \geq \gamma^\top v, \tag{2.20}$$

where the inequality (2.20) holds strictly whenever $u_j - v_j \neq 0$. Now, we use these facts to prove that $f_\gamma(v) \leq f_\gamma(u)$. If u fits in the first clause of (2.6), then $f_\gamma(v) \leq \gamma^\top v + 1 \leq \gamma^\top u + 1 = f_\gamma(u)$, where the second inequality follows from (2.20). Assume u does not satisfies the first clause. If v does not fit in the first clause, then the result follows directly from (2.20) and the fact that $\lceil \cdot \rceil$ is non-decreasing. Suppose v satisfies the first clause, that is $v_j \neq 0$ and $\gamma^\top v \in \mathbb{Z}$. In this case, if $u_j = 0$, then $u_j - v_j \neq 0$ and hence (2.20) holds strictly. Therefore, we conclude that $f_\gamma(v) = \gamma^\top v + 1 \leq \lceil \gamma^\top u \rceil = f_\gamma(u)$. On the other hand, if $u_j \neq 0$, then $\gamma^\top u \notin \mathbb{Z}$ (since u does not satisfy the first clause), and using (2.20) we obtain $\gamma^\top v < \lceil \gamma^\top u \rceil$ and hence $f_\gamma(v) = \gamma^\top v + 1 \leq \lceil \gamma^\top u \rceil = f_\gamma(u)$, which completes the proof. $\square$

2.5 Application of cut-generating functions in $\mathbb{R}^2$

In this section, we will prove Theorem 2.3. We begin with proofs of two technical lemmas.

Lemma 7. *Let $W^i = \{x \in \mathbb{R}^2 \mid A^i x \succeq_{\mathcal{L}^{m_i}} b^i\}$ be a parabola, where $A^i \in \mathbb{R}^{m_i \times 2}$, $b^i \in \mathbb{R}^{m_i}$ and $\mathcal{L}^{m_i}$ is the second-order cone in $\mathbb{R}^{m_i}$. If $\pi \in \text{rec.cone}^*(W^i) \setminus \text{int}\,(\text{rec.cone}^*(W^i))$, $\pi \neq 0$, then the problem $\inf\{\pi^\top x \mid x \in W^i\}$ is unbounded.*

Proof. Up to a rotation, any parabola in $\mathbb{R}^2$ can be written as $\{(x, y) \in \mathbb{R}^2 \mid y \geq \rho(x - x_0)^2 + y_0\}$, where $\rho > 0$. In this case, the recession cone of the parabola is a vertical line. As $\pi \in \text{rec.cone}^*(W^i) \setminus \text{int}\,(\text{rec.cone}^*(W^i))$ we must have $\pi_2 = 0$, in which case $\pi_1 \neq 0$ and the problem is clearly unbounded. $\square$

Lemma 8. *Let W be the set as in the statement of Theorem 2.3. Assume, in addition, that W is unbounded. Let $\pi \neq 0$ be such that $\pi \notin \text{int}\,(\text{rec.cone}^*(W))$. If the problem*

$$\alpha := \inf\{\pi^\top x \mid x \in W\} \tag{2.21}$$

is bounded, then there exists $i_0 \in [m]$ such that

$$\alpha = \inf\{\pi^\top x \mid W^{i_0}\}. \tag{2.22}$$

Moreover, $W^{i_0} = \{x \in \mathbb{R}^2 \mid A^{i_0}x \succeq_{\mathcal{L}^{m_{i_0}}} b^{i_0}\}$ is either:

(i) a half-space defined by $\pi^\top x \geq \alpha$; or

(ii) one branch of a hyperbola whose one of the asymptotes is orthogonal to π.

Proof. Since the primal problem (2.21) is bounded and strictly feasible, we have that its dual

$$\sup\{\sum_{i=1}^{m}(b^i)^\top y^i \mid \sum_{i=1}^{m}(y^i)^\top A^i = \pi^\top,\ y^i \in \mathcal{L}^*_{m_i}\ \forall i \in [m]\} \tag{2.23}$$

is solvable [19]. We will show that (2.23) admits an optimal solution for which $y^i = 0$ for all $i \in [m]$ except for one particular $i_0 \in [m]$.

Since (2.21) is bounded, it follows from Lemma 3 that $\pi \in$ rec.cone$^*(W)$. On the other hand, by assumption π is not in the interior of that cone. Therefore, using Lemma 2 we conclude that there exists a non-zero vector $d_0 \in$ rec.cone(W) such that $\pi^\top d_0 = 0$. Then any feasible solution $(y^1, y^2, \cdots, y^m)$ of (2.23) satisfies

$$0 = \pi^\top d_0 = \sum_{i=1}^{m}(y^i)^\top A^i d_0.$$

Moreover, each term in this summation is non-negative since $A^i d_0 \succeq_{\mathcal{L}_{m_i}} 0$ (recall $d_0 \in$ rec.cone(W)) and $y^i \in \mathcal{L}^*_{m_i}$, for all $i \in [m]$. As a result, we have $(y^i)^\top A^i d_0 = 0\ \forall i \in [m]$. As d_0 is a non-zero vector in $\mathbb{R}^2$, we conclude that for each $i \in [m]$ there must exist a scalar λ_i such that

$$(y^i)^\top A^i = \lambda_i \pi^\top. \tag{2.24}$$

We claim that $\lambda_i \geq 0$ for all $i \in [m]$. To prove the claim, all we need to show is that $(y^i)^\top A^i$ and π are in the same half-space. By assumption $\pi \in$ rec.cone$^*(W)$. Since rec.cone$^*(W)$ is contained in a half-space (otherwise we would have rec.cone$(W) = \{0\}$ which contradicts the fact that W is unbounded), it is enough to prove that $(y^i)^\top A^i \in$ rec.cone$^*(W)$. To see why this is true, note that for all $d \in$ rec.cone(W^i) we have $A^i d \succeq_{\mathcal{L}_{m_i}} 0$, which implies $(y^i)^\top A^i d \geq 0$. Thus, $(y^i)^\top A^i \in$ rec.cone$^*(W^i) \subseteq$ rec.cone$^*(W)$, where the last containment follows from the fact that rec.cone$(W^i) \supseteq$ rec.cone(W).

Now, suppose $(y^1, y^2, \cdots, y^m)$ is an optimal solution of the dual problem (2.23). If $\lambda_i = 0$, then we must have $(b^i)^\top y^i = 0$, because $(b^i)^\top y^i > 0$ would imply the dual problem to be unbounded and $(b^i)^\top y^i < 0$ would imply that the current solution is not optimal. Hence we have that if $\lambda_i = 0$, then we can set $y^i = 0$ without altering the objective value. On the other hand, since $\pi \neq 0$, (2.24) combined with the equality in (2.23) imply that the λ's add up to 1. Thus, we cannot have $\lambda_i = 0$ for all $i \in [m]$. Suppose $\lambda_i, \lambda_j > 0$ for some $i, j \in [m]$, $i \neq j$. We claim that $(b^i)^\top y^i = (\lambda_i/\lambda_j)(b^j)^\top y^j$. Without loss of generality, assume by contradiction that $(b^i)^\top y^i < (\lambda_i/\lambda_j)(b^j)^\top y^j$. Then, since $\lambda_i + \lambda_j \leq 1$ we obtain

$$(b^i)^\top y^i + (b^j)^\top y^j < \frac{\lambda_i}{\lambda_j}(b^j)^\top y^j + (b^j)^\top y^j \leq \frac{1}{\lambda_j}(b^j)^\top y^j.$$

In this case, we could set $\lambda_i = 0,\ \lambda_j = 1$ and $y^i = 0$ to obtain a new feasible solution with objective value strictly larger. But this contradicts the fact that y is an optimal solution. Thus, the claim holds and by setting $\lambda_i = 0, \lambda_j = 1$ and $y^i = 0$ we obtain a new feasible solution with the same objective value, and hence optimal. In this case, we set $i_0 = j$.

Consider now the primal-dual pair

$$\beta := \inf\{\pi^\top x \mid\ A^{i_0} x \succeq_{\mathcal{L}_{m_{i_0}}} b^{i_0}\}, \tag{2.25}$$

$$\sup\{(b^{i_0})^\top y^{i_0} \mid (y^{i_0})^\top A^{i_0} = \pi^\top,\ y^{i_0} \in \mathcal{L}^*_{m_{i_0}}\}. \tag{2.26}$$

Let x^* be an ε-optimal solution to the original primal (2.21), that is $x^* \in W$ and $\pi^\top x^* \leq$

$\alpha + \varepsilon$. Clearly, x^* is feasible for (2.25). Note now that the dual solution constructed above for (2.23), when restricted to the y^{i_0} component is a feasible solution to (2.26) with objective value α. Thus, we have $\alpha \leq \beta \leq \pi^\top x^* \leq \alpha + \varepsilon$, where the first inequality follows from weak duality to the primal-dual pair (2.25-2.26) and the second inequality follows from fisibility of x^* to (2.25). By taking the limit as ε goes to zero, we obtain (2.22).

To prove the second part of the lemma, we first observe that rec.cone$^*(W^{i_0}) \subseteq$ rec.cone$^*(W)$. If $\pi \notin$ rec.cone$^*(W^{i_0})$, then (2.22) would be unbounded by Lemma 3. As $\pi \notin \mathrm{int}\,($rec.cone$^*(W))$, we have that $\pi \notin \mathrm{int}\,($rec.cone$^*(W^{i_0}))$. Hence, $\pi \in$ rec.cone$^*(W^{i_0}) \setminus \mathrm{int}\,($rec.cone$^*(W^{i_0}))$.

Now, W^{i_0} cannot define an ellipse because then $W \subseteq W^{i_0}$ would be bounded. Since $\pi \in$ rec.cone$^*(W^{i_0}) \setminus$ interior(rec.cone$^*(W^{i_0})$), if W^{i_0} was a parabola, then problem (2.22) would be unbounded in view of Lemma 7. Therefore, only two possibilities remain:
(i) W^{i_0} is defined by a linear inequality, say $\mu^\top x \geq \mu_0$. In this case μ must be a multiple of π, otherwise problem (2.22) would be unbounded. Thus, we may assume $\pi = \mu$ and then $\mu_0 = \alpha$.
(ii) W^{i_0} is one branch of a hyperbola. In this case, rec.cone(W^{i_0}) is defined by the asymptotes of the hyperbola. As $\pi \in$ rec.cone$^*(W^{i_0}) \setminus$ interior(rec.cone$^*(W^{i_0})$), π must be orthogonal to one of the asymptotes. □

Next we prove Lemma 1 that was stated in Section 2.2.3.

Proof. (of Lemma 1) Any conic section (parabola, ellipse, hyperbola) in $\mathbb{R}^2$ is a curve defined by a quadratic equation of the form

$$\frac{1}{2}x^\top Qx + d^\top x + s = 0, \tag{2.27}$$

where s is a scalar, $d \in \mathbb{R}^2$ and $Q = VDV^\top$. In this factorization, $V \in \mathbb{R}^{2\times 2}$ is orthonormal and

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix},$$

where λ_1, λ_2 are the eigenvalues of Q. In particular, the curve defined by (2.27) is a hyperbola if and only if one of these eigenvalues is positive and the other is negative. After changing variables $y := V'x$ and completing squares, equation (2.27) can be written in exactly one of the following forms

$$[\beta_1(y_1 - \alpha_1)]^2 - [\beta_2(y_2 - \alpha_2)]^2 = \pm\eta^2, \tag{2.28}$$

where η and α_i, β_i, for $i = 1, 2$, are constants depending on the coefficients of (2.27). In what follows, we assume that the coefficient of η^2 is positive. If it was negative, then we could multiply (2.28) by -1 and all we will do next would be analogous. Under this assumption, one branch of the hyperbola is given by

$$\begin{aligned} G^+ &:= \{y \in \mathbb{R}^2 \mid (\eta)^2 + [\beta_2(y_2 - \alpha_2)]^2 \leq [\beta_1(y_1 - \alpha_1)]^2,\ \beta_1(y_1 - \alpha_1) \geq 0\} \\ &= \{y \in \mathbb{R}^2 \mid \sqrt{\eta^2 + [\beta_2(y_2 - \alpha_2)]^2} \leq \beta_1(y_1 - \alpha_1)\} \\ &= \{y \in \mathbb{R}^2 \mid (\eta, \beta_2(y_2 - \alpha_2), \beta_1(y_1 - \alpha_1)) \in \mathcal{L}^3\} \\ &= \{y \in \mathbb{R}^2 \mid \begin{bmatrix} 0 & 0 \\ 0 & \beta_2 \\ \beta_1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \succeq_{\mathcal{L}^3} \begin{bmatrix} -\eta \\ \beta_2\alpha_2 \\ \beta_1\alpha_1 \end{bmatrix} \}. \end{aligned}$$

Then, going back to the space of the original variables we obtain

$$G^+ = \{x \in \mathbb{R}^2 \mid \begin{bmatrix} 0 & 0 \\ \beta_2 v_{12} & \beta_2 v_{22} \\ \beta_1 v_{11} & \beta_1 v_{21} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \succeq_{\mathcal{L}^3} \begin{bmatrix} -\eta \\ \beta_2\alpha_2 \\ \beta_1\alpha_1 \end{bmatrix} \},$$

where v_{ij} are the entries of the matrix V. The other branch of the hyperbola is given by

$$G^- := \{y \in \mathbb{R}^2 \,|\, (\eta)^2 + [\beta_2(y_2 - \alpha_2)]^2 \leq [\beta_1(y_1 - \alpha_1)]^2,\ \beta_1(y_1 - \alpha_1) \leq 0\}.$$

After the change of variables $\tilde{y} := -y$ we obtain

$$\begin{aligned}
G^- &= \{\tilde{y} \in \mathbb{R}^2 \,|\, (\eta)^2 + [\beta_2(-\tilde{y}_2 - \alpha_2)]^2 \leq [\beta_1(-\tilde{y}_1 - \alpha_1)]^2,\ \beta_1(-\tilde{y}_1 - \alpha_1) \leq 0\} \\
&= \{\tilde{y} \in \mathbb{R}^2 \,|\, (\eta)^2 + [\beta_2(\tilde{y}_2 + \alpha_2)]^2 \leq [\beta_1(\tilde{y}_1 + \alpha_1)]^2,\ \beta_1(\tilde{y}_1 + \alpha_1) \geq 0\} \\
&= \{\tilde{y} \in \mathbb{R}^2 \,|\, (\eta, \beta_2(\tilde{y}_2 + \alpha_2), \beta_1(\tilde{y}_1 + \alpha_1)) \in \mathcal{L}^3\} \\
&= \{\tilde{y} \in \mathbb{R}^2 \,|\, \begin{bmatrix} 0 & 0 \\ 0 & \beta_2 \\ \beta_1 & 0 \end{bmatrix} \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \end{bmatrix} \succeq_{\mathcal{L}^3} \begin{bmatrix} -\eta \\ -\beta_2\alpha_2 \\ -\beta_1\alpha_1 \end{bmatrix} \}.
\end{aligned}$$

Going back to the space of the original variables we obtain

$$G^- = \{x \in \mathbb{R}^2 \,|\, \begin{bmatrix} 0 & 0 \\ -\beta_2 v_{12} & -\beta_2 v_{22} \\ -\beta_1 v_{11} & -\beta_1 v_{21} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \succeq_{\mathcal{L}^3} \begin{bmatrix} -\eta \\ -\beta_2\alpha_2 \\ -\beta_1\alpha_1 \end{bmatrix} \}.$$

It follows from (2.28) that the asymptotes of G^+ have equations

$$\begin{aligned}
\beta_1 y_1 + \beta_2 y_2 &= \beta_1\alpha_1 + \beta_2\alpha_2, \\
\beta_1 y_1 - \beta_2 y_2 &= \beta_1\alpha_1 - \beta_2\alpha_2.
\end{aligned}$$

In the space of x variables they become

$$\begin{aligned}
(\beta_1 v_{11} + \beta_2 v_{12})x_1 + (\beta_1 v_{21} + \beta_2 v_{22})x_2 &= \beta_1\alpha_1 + \beta_2\alpha_2, \\
(\beta_1 v_{11} - \beta_2 v_{12})x_1 + (\beta_1 v_{21} - \beta_2 v_{22})x_2 &= \beta_1\alpha_1 - \beta_2\alpha_2.
\end{aligned} \tag{2.29}$$

The asymptotes of G^- are obtained in a similar way. $\square$

Lemma 9. *Let G be one branch of a non-degenerate hyperbola in $\mathbb{R}^2$. Let $\pi^\top x \geq \pi_0$ be a face of G^I such that $\pi \in \mathbb{Z}^2$ is non-zero and orthogonal to one of the asymptotes. Then $\pi^\top x \geq \pi_0$ is one of the inequalities (2.13).*

Proof. Using the same notation adopted in the proof of Lemma 1 above, we assume $G = G^+$. If $G = G^-$, then the proof is analogous. Note that G is contained in the set

$$\begin{aligned} H := \{x \in \mathbb{R}^2 \mid\, & (\beta_1 v_{11} + \beta_2 v_{12})x_1 + (\beta_1 v_{21} + \beta_2 v_{22})x_2 \geq \beta_1\alpha_1 + \beta_2\alpha_2, \\ & (\beta_1 v_{11} - \beta_2 v_{12})x_1 + (\beta_1 v_{21} - \beta_2 v_{22})x_2 \geq \beta_1\alpha_1 - \beta_2\alpha_2\}. \end{aligned}$$

Assume π is orthogonal to the asymptote (2.29). The proof of the case in which π is orthogonal to the second asymptote is similar. Since $\pi \in \mathbb{Z}^2$ is non-zero, we may assume that the coefficients of x_1 and x_2 in (2.29) are integers. Let

$$\tau := \gcd\{\beta_1 v_{11} + \beta_2 v_{12}, \beta_1 v_{21} + \beta_2 v_{22}\}.$$

Since the hyperbola is non-degenerate, the line

$$(\beta_1 v_{11} + \beta_2 v_{12})x_1 + (\beta_1 v_{21} + \beta_2 v_{22})x_2 = \beta_1\alpha_1 + \beta_2\alpha_2$$

does not intersect G. However, for all $\varepsilon > 0$, the equation

$$\frac{\beta_1 v_{11} + \beta_2 v_{12}}{\tau}x_1 + \frac{\beta_1 v_{21} + \beta_2 v_{22}}{\tau}x_2 = \frac{\beta_1\alpha_1 + \beta_2\alpha_2}{\tau} + \varepsilon \tag{2.30}$$

intersects G along a ray. Moreover, (2.30) has integral solutions if and only if the right-hand-side is integral.

Therefore, if $(\beta_1\alpha_1 + \beta_2\alpha_2)/\tau \in \mathbb{Z}$, then the inequality

$$\frac{\beta_1 v_{11} + \beta_2 v_{12}}{\tau} x_1 + \frac{\beta_1 v_{21} + \beta_2 v_{22}}{\tau} x_2 \geq \frac{\beta_1\alpha_1 + \beta_2\alpha_2}{\tau} + 1 \tag{2.31}$$

is a face of G^I, and hence it is equivalent to $\pi^\top x \geq \pi_0$. On the other hand, if $(\beta_1\alpha_1 + \beta_2\alpha_2)/\tau \notin \mathbb{Z}$, then

$$\frac{\beta_1 v_{11} + \beta_2 v_{12}}{\tau} x_1 + \frac{\beta_1 v_{21} + \beta_2 v_{22}}{\tau} x_2 \geq \left\lceil \frac{\beta_1\alpha_1 + \beta_2\alpha_2}{\tau} \right\rceil \tag{2.32}$$

is a face of G^I, and hence it is equivalent to $\pi^\top x \geq \pi_0$.

Observe now that (2.31) and (2.32) are one of the inequalities (2.13) in view of Proposition 1. □

Next we use Lemma 8 and Lemma 9 above to proof Theorem 2.3.

Proof. *(of Theorem 2.3)* First, we observe that if W is bounded, then the result follows directly from Theorem 2.1. Suppose W is unbounded. We have two cases:

Case 1: $W \cap \mathbb{Z}^2 = \emptyset$. In this case, there exist $\pi = (\pi_1, \pi_2)$ with π_1, π_2 integer relatively prime and a integer π_0 such that [56, 15]

$$W \subseteq \{x \in \mathbb{R}^2 \mid \pi_0 \leq \pi^\top x \leq \pi_0 + 1\}. \tag{2.33}$$

We will show that the cut $\pi^\top x \geq \pi_0 + 1$ can be obtained using subadditive functions (2.3) or using one of the inequalities (2.13). Analogous proof holds for the cut $\pi^\top x \leq \pi_0$. A consequence of W being between these two lines is that rec.cone(W) is orthogonal to π and, therefore, $\pi \notin \operatorname{int}(\text{rec.cone}^*(W))$ in view of Lemma 2. Then, by Lemma 8,

$$\alpha := \inf\{\pi^\top x \mid W^{i_0}\} = \inf\{\pi^\top x \mid x \in W\},$$

for some $i_0 \in [m]$, where there are only two possibilities for $W^{i_0} = \{x \in \mathbb{R}^2 \mid A^{i_0} x \succeq_{\mathcal{L}^{m_{i_0}}}$

$b^{i_0}\}$:

(i) W^{i_0} is the half-space $\pi^\top x \geq \alpha$: In this case, since $A^{i_0}x \succeq_{\mathcal{L}^{m_{i_0}}} b^{i_0}$ is non-redundant, we have that the line $\pi^\top x = \alpha$ intersects W. Note that $\pi_0 \leq \alpha$ in view of (2.33). Since W is unbounded and its recession cone is orthogonal to π, if $\alpha = \pi_0$, then W would contain a integer point from the line $\pi^\top x = \pi_0$. Therefore, $\alpha > \pi_0$ in which case $\pi^\top x \geq \lceil \alpha \rceil = \pi_0 + 1$ is a valid inequality for W^I and this cut can be obtained using a subadditive function (2.3).

(ii) W^{i_0} is a hyperbola whose one of the asymptotes is orthogonal to π: Without loss of generality, we may assume that the asymptote orthogonal to π has equation $\pi^\top x = \alpha$. Let

$$\beta = \begin{cases} \alpha + 1 \text{ if } \alpha \in \mathbb{Z} \\ \lceil \alpha \rceil \text{ if } \alpha \notin \mathbb{Z}. \end{cases} \tag{2.34}$$

Since the hyperbola is non-degenerate, we have that $\pi^\top x \geq \beta$ is a valid inequality for $(W^{i_0})^I$. Moreover, $\pi^\top x = \beta$ contains a ray of W^{i_0} since $\beta > \alpha$. Then, since π_1 and π_2 are relatively prime and $\beta \in \mathbb{Z}$, we have that $\pi^\top x \geq \beta$ is, in addition, a face of $(W^{i_0})^I$. Now, it follows from Lemma 9 that this face is one of the inequalities (2.13). Finally, note that $\pi_0 \leq \alpha < \pi_0 + 1$. Thus, we have that $\beta = \pi_0 + 1$.

Case 2: interior$(W) \cap \mathbb{Z}^2 \neq \emptyset$. By assumption, the components of π are integers and, without loss of generality, we may also assume they are relatively prime. We now have three cases.

1. $\pi \notin$ rec.cone$^*(W)$: In this case, by Lemma 3, we have that $\inf\{\pi^\top x \mid x \in W\}$ is unbounded. Since we assume that interior$(W) \cap \mathbb{Z}^2 \neq \emptyset$, we obtain that $\inf\{\pi^\top x \mid x \in W \cap \mathbb{Z}^2\}$ is unbounded [53], which contradicts the fact that $\pi^\top x \geq \pi_0$ is a valid inequality for W^I.

2. $\pi \in$ interior(rec.cone$^*(W)$) : In this case, $\{x \in W \mid \pi^\top x \leq \pi_0\}$ is bounded in view of Proposition 2. Therefore, it follows from Theorem 2.1 that the valid inequality $\pi^\top x \geq \pi_0$ can be obtained using functions (2.3).

3. $\pi \in$ rec.cone$^*(W) \setminus$ interior(rec.cone$^*(W)$): Since interior$(W) \cap \mathbb{Z}^2 \neq \emptyset$ and $\inf\{\pi^\top x \mid x \in W \cap \mathbb{Z}^2\}$ is bounded, we have that $\alpha := \inf\{\pi^\top x \mid x \in W\}$ is bounded [53]. Then, by Lemma 8, $\alpha = \inf\{\pi^\top x \mid W^{i_0}\}$, for some $i_0 \in [m]$, where there are only two possibilities for $W^{i_0} = \{x \in \mathbb{R}^2 \mid A^{i_0} x \succeq_{\mathcal{L}^{m_{i_0}}} b^{i_0}\}$:

(i) W^{i_0} is the half-space $\pi^\top x \geq \alpha$: Since $A^{i_0} x \succeq_{\mathcal{L}^{m_{i_0}}} b^{i_0}$ is non-redundant, we have that the line $\pi^\top x = \alpha$ intersects W. Thus, $\pi^\top x \geq \lceil \alpha \rceil$ is a valid inequality for W^I and this cut can be obtained using a subadditive function (2.3). Now, we only need to show that $\lceil \alpha \rceil = \pi_0$. It is enough to show that the line $\pi^\top x = \lceil \alpha \rceil$ intersects $W \cap \mathbb{Z}^2$. Note that the line $\pi^\top x = \lceil \alpha \rceil$ intersects W (otherwise we would have $W \subseteq \{x \in \mathbb{R}^2 \mid \pi^\top x < \lceil \alpha \rceil\}$ which contradicts the fact that $W \cap \mathbb{Z}^2 \neq \emptyset$ since $\pi^\top x \geq \lceil \alpha \rceil$ is valid inequality for W^I). Thus, $\{x \in W \mid \pi^\top x = \lceil \alpha \rceil\} \neq \emptyset$. Moreover, since $\pi \in$ rec.cone$^*(W) \setminus$ interior(rec.cone$^*(W)$), there exists a non-zero vector $d \in$ rec.cone(W) such that $\pi^\top d = 0$. Therefore, d is in the recession cone of $\{x \in W \mid \pi^\top x = \lceil \alpha \rceil\}$. Hence, $\pi^\top x = \lceil \alpha \rceil$ contains a ray of W. Thus, $\pi^\top x = \lceil \alpha \rceil$ contains an integer point of W since π_1 and π_2 are relatively prime.

(ii) W^{i_0} is a hyperbola one of whose asymptotes is orthogonal to π: As in Case 1 *(ii)*, we can show that $\pi^\top x \geq \beta$ is a face of W^{i_0}, where β is defined in (2.34). Moreover, by Lemma 9, $\pi^\top x \geq \beta$ is one of the inequalities (2.13). Now, only remains to show that $\beta = \pi_0$. It is enough to show that $\pi^\top x = \beta$ intersects $W \cap \mathbb{Z}^2$. Clearly, $\pi^\top x \geq \beta$ is a valid inequality for $W^I \subseteq W^{i_0}$. Since $\alpha < \beta$, we have that the line $\pi^\top x = \beta$ intersects W (otherwise we would have $W \subseteq \{x \in \mathbb{R}^2 \mid \pi^\top x < \beta\}$ which contradicts the fact that $W \cap \mathbb{Z}^2 \neq \emptyset$). Therefore, as in the case *(i)* above, we can prove that $\pi^\top x = \beta$ contains a ray of W. Thus, $\pi^\top x = \beta$ contains an integer point of W since π_1 and π_2 are relatively prime and $\beta \in \mathbb{Z}$.

□

CHAPTER 3

NEW SOCP RELAXATION AND BRANCHING RULE FOR BIPARTITE BILINEAR PROGRAMS

The work presented in this chapter has already been published [55].

3.1 Introduction: Bipartite bilinear program (BBP)

A quadratically constrained quadratic program (QCQP) is called as a bilinear optimization problem if every degree two term in the constraints and objective involves the product of two distinct variables. For a given instance of bilinear optimization problem, one often associates a simple graph constructed as follows: The set of vertices corresponds to the variables in the instance and there is an edge between two vertices if there is a degree two term involving the corresponding variables in the instance formulation. Strength of various convex relaxations for bilinear optimization problems can be analyzed using combinatorial properties of this graph [92, 27, 68].

When this graph is bipartite, we call the resulting bilinear problem as a bipartite bilinear program (BBP). In other words, BBP is an optimization problem of the following form:

$$\begin{aligned} \min \quad & x^\top Q_0 y + d_1^\top x + d_2^\top y \\ \text{s.t.} \quad & x^\top Q_k y + a_k^\top x + b_k^\top y + c_k = 0, \ k \in \{1, \dots, m\} \\ & l \leq (x, y) \leq u \\ & (x, y) \in \mathbb{R}^{n_1 + n_2}, \end{aligned} \tag{3.1}$$

where $n_1, n_2 \in \mathbb{Z}_+, \ Q_0, Q_k \in \mathbb{R}^{n_1 \times n_2}, \ d_1, a_k \in \mathbb{R}^{n_1}, \ d_2, b_k \in \mathbb{R}^{n_2}, \ c_k \in \mathbb{R}, \forall k \in \{1, \dots, m\}$. The vectors $l, u \in \mathbb{R}^{n_1+n_2}$ define the box constraints on the decision variables and, without loss of generality, we assume that $l_i = 0, \ u_i = 1, \ \forall i \in \{1, \dots, n_1 + n_2\}$.

BBP (3.1) may include bipartite bilinear inequality constraints, which can be converted into equality constraints by adding slack variables, and these slack variables will also be bounded since the original variables are bounded. Also notice that a squared term x_i^2 can be converted into a bipartite bilinear form by replacing it with the product of two new variables toguether with a linear equation establishing that these two new variables are equal.

We note that BBP is a special case of the more general biconvex optimization problem [65]. BBP has many applications such as waste water management [58, 38, 60], pooling problem [67, 71], and supply chain [107].

3.2 Our results

For an integer $n \geq 1$, we use $[n]$ to describe the set $\{1, \dots, n\}$.

3.2.1 Second-order cone representable relaxation of BBP

A common and successful approach in integer linear programing is to generate cutting-planes implied by single constraint relaxation, see for example [45, 94, 52, 25]. We take a similar approach here. We begin by examining one row relaxation of BBP, that is, we study the convex hull of the set defined by a single constraint defining the feasible region of (3.1). Our first result is to show that the convex hull of this set is second-order cone representable (SOCr) in the extended space, where we have introduced new variables w_{ij} for $x_i y_j$. We formally present this result next.

Theorem 3.1. *Let $n_1, n_2 \in \mathbb{Z}_+$, $V_1 \in [n_1]$, $V_2 \in [n_2]$, and $E \subseteq V_1 \times V_2$. Consider the one-constraint BBP set*

$$S := \left\{ (x, y, w) \in [0,1]^{n_1+n_2+|E|} \;\middle|\; \begin{array}{l} \displaystyle\sum_{(i,j)\in E} q_{ij} w_{ij} + \sum_{i \in V_1} a_i x_i + \sum_{j \in V_2} b_j y_j + c = 0, \\ w_{ij} = x_i y_j, \ \forall (i,j) \in E \end{array} \right\}.$$

Then:

(i) Let $(\bar{x}, \bar{y}, \bar{w})$ be an extreme point of S. Then, there exists $U \subseteq V_1 \cup V_2$, of the form

(a) $U = \{i_0, j_0\}$ where $(i_0, j_0) \in E$, or

(b) $U = \{i_0\}$ where $i_0 \in V_1$ is an isolated node, or

(c) $U = \{j_0\}$ where $j_0 \in V_2$ is an isolated node,

such that $\bar{x}_i \in \{0, 1\},\ \forall i \in V_1 \setminus U$, and $\bar{y}_j \in \{0, 1\},\ \forall j \in V_2 \setminus U$.

(ii) $\text{conv}(S)$ *is SOCr.*

A proof of Theorem 3.1 is presented in Section 3.3.1.

Remark 3. *In Theorem 3.1, part (ii) follows from part (i). For any given choice of U, we first fix all the variables to 0 or 1 except for those in U. It is then shown that the convex hull of the resulting set is SOCr and we obtain (ii) by convexifying the union of a finite set of SOCr sets.*

It is easy to see that the number of distinct U sets is $\mathcal{O}(n_1 n_2)$, and the number of possible fixings is $\mathcal{O}(2^{n_1+n_2})$. Thus, the number of resulting SOCr objects is $\mathcal{O}(n_1 n_2 2^{n_1+n_2})$.

We note that the literature in global optimization theory has many results on convexifying functions, see for example [4, 118, 99, 136, 137]. However, as is well-known, replacing a constraint $f(x) = b$ by $\{x \,|\, \hat{f}(x) \geq b,\ \breve{f}(x) \leq b\}$ where $\hat{f}$ and $\breve{f}$ are the concave and convex envelop of f, does not necessarily yield the convex hull of the set $\{x \,|\, f(x) = b\}$. There are relatively lesser number of results on convexification of sets [135, 110, 111, 134]. Theorem 3.1 generalizes results presented in [134, 66, 82] and is related to results presented in [48].

The SOCP relaxation for the feasible region of the general BBP (3.1) that we propose, henceforth referred as S^{SOCP}, is the intersection of the convex hull of each of the constraints of (3.1). Formally:

$$S^{SOCP} = \bigcap_{k=1}^{m} \text{conv}(S_k),$$

where $S_k = \{(x, y, w) \in [0,1]^{n_1 \times n_2 \times |E|} \mid x^\top Q_k y + a_k^\top x + b_k^\top y + c_k = 0, w_{ij} = x_i y_j \; \forall (i,j) \in E\}$ and E is the edge set of the graph corresponding to the BBP instance (and not just of one row). As an aside, note that S^{SOCP} can be further strengthened by adding the convex hull of single row BBP sets arrived by taking linear combinations of rows.

Next, we discuss the strength of S^{SOCP} vis-á-vis the strength of other standard relaxations. Consider the following two standard relaxations of the feasible region of BBP (3.1): Let S^{SDP} be the standard semidefinite programming (SDP) relaxation (see (3.14-3.16) for precise definition), and let

$$\begin{aligned} S^{QBP} \;\; := \;\; & \left\{ (x, y, w) \in [0,1]^{n_1+n_2+|E|} \mid \sum_{(ij) \in E} (Q_k)_{ij} w_{ij} + a_k^\top x + b_k^\top y + c_k = 0 \; k \in [m] \right\} \\ & \bigcap \text{conv}\left(\{(x, y, w) \in [0,1]^{n_1+n_2+|E|} \mid w_{ij} = x_i y_j \; \forall (i,j) \in E\} \right). \end{aligned} \tag{3.2}$$

Note that S^{QBP} is a polyhedral set, since the second set in the right-hand-side of (3.2) is equal to the Boolean Quadratic Polytope [36]. Two well-known classes of valid inequalities for this set are the McCormick's inequalities [4] and the triangle inequalities [114].

Theorem 3.2. *For any BBP, we have that*

$$\text{proj}_{x,y,w}\left(S^{SDP}\right) \bigcap S^{QBP} \supseteq S^{SOCP}.$$

A proof of Theorem 3.2 is presented in Section 3.3.2. We remark that the containment in Theorem 3.1 can be strict. For instance, consider the set $S := \{(x, y) \in [0,1]^2 \mid 10xy = 1\}$, which is represented by the thick curve in Figure 3.1. The same figure illustrates how S^{SOCP}, the darkest region, is strictly contained in $\text{proj}_{x,y}\left(S^{SDP}\right) \bigcap \text{proj}_{x,y}\left(S^{QBP}\right)$. Notice that S^{QBP} coincides with the McCormick relaxation in this simple case, and $\text{proj}_{x,y,w}\left(S^{SDP}\right)$ is strictly contained in S^{QBP}.

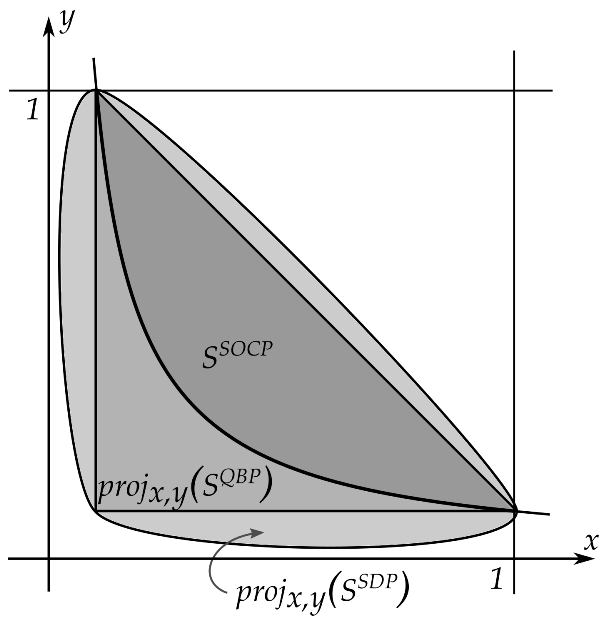

Figure 3.1: Illustration of strict containtment in Theorem 3.2.

Remark 4. *It is possible to show that the convex hull of one row BBP is SOCr, even without introducing the w variables. Thus, it is possible to construct, similar to S^{SOCP}, a SOCr relaxation of BBP, without introducing w variables. However, this SOCP relaxation would be weaker. In particular, we are unable to prove the corresponding version of Theorem 3.2 for this SOCP relaxation. The strength of S^{SOCP} relaxation is due to the fact that the extended space w variables 'interact' from different constraints. For example, consider the two-dimensional BBP set defined a follows:*

$$
\begin{aligned}
S &= \{(x,y) \in [0,1]^2 \,|\, 10xy = 1,\ 10(x-1)(y-1) = 1\} \\
&= \{(x,y) \in [0,1]^2 \,|\, 10xy = 1,\ 10xy - 10x - 10y = -9\}.
\end{aligned}
$$

Points A and B in Figure 3.2 represent the only two feasible solutions of S. Therefore, the convex hull of S is the line segment AB. The shaded region represents S^{SOCP} in the space *of x and y, i.e., the intersection of the convex hull of the two equations over the $0-1$ box. However, when we write S^{SOCP} in the extended space of $w = xy$, we obtain the implied equation $x+y=1$ (implied by $10w = 1$ and $10w - 10x - 10y = -9$) and then the exact convex hull of S.*

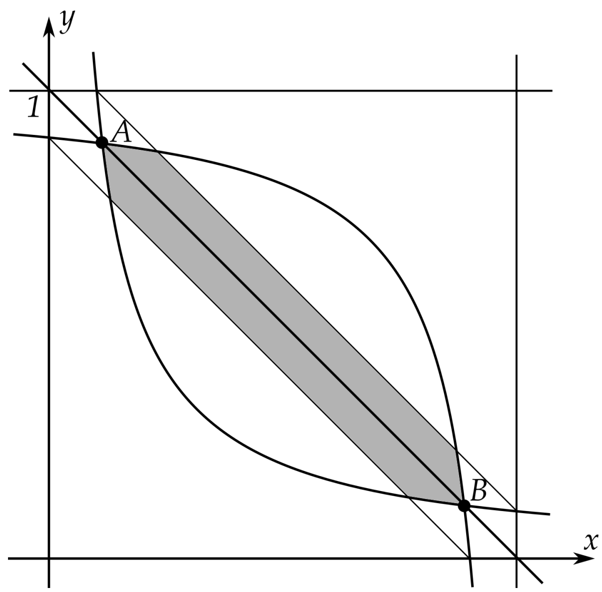

Figure 3.2: Illustration of a S^{SOCP} relaxation.

We note that other SOCP relaxations for QCQPs have been proposed [81, 35]. However, these are all weaker than the standard SDP relaxation.

We also note that it is polynomial time to optimize on S^{SDP}, although the tractability of solving SDPs in practice is still limited. On the other hand, solvers for SOCP are significantly better in practice. It is NP-hard to optimize on S^{QBP}, although as discussed in Remark 3, the size of the extended formulation to obtain S^{SOCP} is exponential in size.

3.2.2 A new branching rule

For details about general branch-and-bound scheme for global optimization see, for example, [119]. Inspired by the convex relaxation described in Section 3.2.1, we propose a new rule for partitioning the domain of a given variable in order to produce two branches. Details of this new proposed branching rule together with node selection and variable selection rules that we used in our computational experiments are presented in Section 3.4.

Here, we sketch the main ideas behind our new proposed branching rule. Suppose we have decided to branch on the variable x_1. As explained in Remark 3, the convex hull of the one constraint set is obtained by taking the convex hull of union of sets obtained by fixing all but two (or one) variables. If we are branching on x_1, we examine all such two-variable sets involving x_1 obtained from each of the constraints. For each of these sets, there is an ideal point to divide the range of x_1 so that the sum of the volume of the two convex hulls of the two-dimensional sets corresponding to the two resulting branches is minimized. (See recent papers on importance of volume minimization in branch-and-bound algorithm [130]). We present a heuristic to find an "ideal range". We collect all such ideal ranges corresponding to all the two-dimensional sets involving x_1. Then we present a heuristic to select one points (based on corresponding volume reduction) to finally partition the domain of x_1. We also use similar arguments to propose a new variable selection rule.

3.2.3 A new application of BBP and computational experiments

A new application of BBP, which motivated our work presented here, is called as the *finite element model updating problem*, which is a fundamental methodological problem in structural engineering. See Section 3.5.1 for a description of the problem. All the new methods we develop here are tested on instances of this problem.

Due to the large size of S^{SOCP}, in practice, we consider a lighter version of this relaxation. In particular, we write the extended formulation of each row of BBP corresponding only to the variables in that row (see details in Section 3.5.2). As our instances are row sparse, the resulting SOCP relaxation can be solved in reasonable time. Unfortunately, there are no theoretical guarantees for the bounds of this light version of the relaxation. After some preliminary experimentation, we observed that a polyhedral outer approximation of the SOCP relaxation produces similar bounds but solves much faster. Therefore, we used this linear programming (LP) relaxation in our experiments. Details of this outer approximation is presented in Section 3.5.2.

Our computational experiments are aimed at making three comparisons. First, we examined the quality of the dual bound produced at root node via our new method (polyhedral outer approximation of SOCP relaxation) against SDP, McCormick, and SDP together with McCormick inequalities. The bounds produced are better for the new method. Second, we test the performance of the new branching rule against traditional branching rules. Our experiments show that the new branching rule significantly out performs the other branching rules. Finally, we compare the performance of our naive branch-and-bound implementation against BARON. In all instances, we close significantly more gap in equal amount of time. All these results are discussed in detail in Section 3.5.3.

3.3 Second-order cone representable relaxation and its strength

3.3.1 Proof of Theorem 3.1

Consider the bipartite graph $G = (V_1, V_2, E)$ defined by the set of vertices $V_1 = [n_1]$ and $V_2 = [n_2]$ which is associated to the equation

$$\sum_{(i,j)\in E} q_{ij}x_iy_j + \sum_{i\in V_1} a_ix_i + \sum_{j\in V_2} b_jy_j + c = 0. \tag{EQ}$$

In this section, we prove that the convex hull of the set

$$S = \{(x, y, w) \in [0, 1]^{n_1+n_2+|E|} \mid (EQ),\ w_{ij} = x_iy_j\ \forall(i, j) \in E\}. \tag{3.3}$$

is SOCr. In addition, the proof provides an implementable procedure to obtain $\mathrm{conv}(S)$. The key idea underlying this result is the fact that, at each extreme point of S, at most two variables are not fixed to 0 or 1 and, once all variables but two (or one) are fixed, the convex hull of the resulting object is SOCr in $\mathbb{R}^2$ (or $\mathbb{R}$). Hence, $\mathrm{conv}(S)$ can be written as the convex hull of an union of SOCr sets.

Preliminary results

First we present a few preliminary results that will be used to prove that $\mathrm{conv}(S)$ is SOCr.

Lemma 10. *[133] Let $f : [0,1]^n \to \mathbb{R}$ be a continuous function and $B \subseteq [0,1]^n$ be a convex set. Then*

$$\mathrm{conv}(\{x \in B \,|\, f(x) = 0\}) = \mathrm{conv}\left(\{x \in B \,|\, f(x) \leq 0\}\right) \bigcap \mathrm{conv}(\{x \in B \,|\, f(x) \geq 0\}).$$

Lemma 11. *[72] Let $f : [0,1]^n \to \mathbb{R}$ be a convex function. Then*

$$G := \mathrm{conv}(\{x \in [0,1]^n \,|\, f(x) \geq 0\}),$$

is a polytope. Indeed, G can be obtained as the convex hull of finite number of points obtained as follows: fix all but one variable to 0 or 1 and solve for $f(x) = 0$.

Lemma 12. *[19] Let $T \subset \mathbb{R}^n$ be a compact set and $\{T_k\}_{k \in K}$ be a partition of the set of all extreme points of T. Then,*

$$\mathrm{conv}(T) = \mathrm{conv}\left(\bigcup_{k \in K} T_k\right) = \mathrm{conv}\left(\bigcup_{k \in K} \mathrm{conv}(T_k)\right). \tag{3.4}$$

In addition, if $\mathrm{conv}(T_k)$ is a SOCr set for every $k \in K$, then $\mathrm{conv}(T)$ is also a SOCr set.

Lemma 13. *Let $B = \{(x,w) \in [0,1]^n \times \mathbb{R} \,|\, x \in B_0,\ w = l^\top x + l_0\}$, where $B_0 \subseteq \mathbb{R}^n$, and $l^\top x + l_0$ is an affine function of x. Then,*

$$\mathrm{conv}(B) = \{(x,w) \in [0,1]^n \times \mathbb{R} \,|\, x \in \mathrm{conv}(B_0),\ w = l^\top x + l_0\}.$$

Proof. We assume B_0 is non-empty, otherwise, there is nothing to prove. Let $(x,w) \in \mathrm{conv}(B)$. Then there exist $(x^i, w^i) \in B$ and $\lambda_i \geq 0,\ \forall i \in [n+2]$, such that $\sum_{i=1}^{n+2} \lambda_i = 1$,

$x = \sum_{i=1}^{n+2} \lambda_i x^i$ and $w = \sum_{i=1}^{n+2} \lambda_i w^i$. It follows by the definition of B that $x^i \in B_0,\ \forall i \in [n+2]$, and hence $x \in \text{conv}(B_0)$. It also follows from the definition of B that $w^i = l^\top x^i + l_0,\ \forall i \in [n+2]$, and hence

$$w = \sum_{i=1}^{n+2} \lambda_i w^i = \sum_{i=1}^{n+2} \lambda_i (l^\top x^i + l_0) = l^\top \left(\sum_{i=1}^{n+2} \lambda_i x^i \right) + l_0 = l^\top x + l_0.$$

Conversely, let (x, w) be such that $x \in \text{conv}(B_0)$ and $w = l^\top x + l_0$. Then, there exist $x^i \in B_0$ and $\lambda_i \geq 0,\ \forall i \in [n+1]$, such that $\sum_{i=1}^{n+1} \lambda_i = 1$, $x = \sum_{i=1}^{n+1} \lambda_i x^i$. Define $w^i = l^\top x^i + l_0,\ \forall i \in [n+1]$. Then $(x^i, w^i) \in B,\ \forall i \in [n+1]$. In addition,

$$w = l^\top x + l_0 = l^\top \left(\sum_{i=1}^{n+1} \lambda_i x^i \right) + l_0 = \sum_{i=1}^{n+1} \lambda_i (l^\top x^i + l_0) = \sum_{i=1}^{n+1} \lambda_i w^i,$$

which completes the proof. □

Proof of part (i) of Theorem 3.1

We restate part (i) of Theorem 3.1 next for easy reference:

Proposition 6. *Let $(\bar{x}, \bar{y}, \bar{w})$ be an extreme point of the set S defined in (3.3). Then, there exists $U \subseteq V_1 \cup V_2$, of the form*

1. *$U = \{i_0, j_0\}$ where $(i_0, j_0) \in E$, or,*
2. *$U = \{i_0\}$ where $i_0 \in V_1$ is an isolated node, or,*
3. *$U = \{j_0\}$ where $j_0 \in V_2$ is an isolated node,*

such that $\bar{x}_i \in \{0, 1\},\ \forall i \in V_1 \setminus U$, and $\bar{y}_j,\ \forall j \in V_2 \setminus U$.

Proof. To prove by contradiction, suppose without loss of generality that $0 < \bar{x}_1, \bar{x}_2 < 1$.

Consider the system of equations

$$\begin{aligned}\bar{a}_1 x_1 + \bar{a}_2 x_2 + \bar{c} &= 0,\\ w_{1j} - x_1 \bar{y}_j &= 0\ \forall j : (1,j) \in E\\ w_{2j} - x_2 \bar{y}_j &= 0\ \forall j : (2,j) \in E,\end{aligned}$$

obtained by fixing $x_i = \bar{x}_i,\ y_j = \bar{y}_j$ in (3.3), $w_{ij} = \bar{x}_i \bar{y}_j\ \forall i \in V_1 \setminus \{1,2\},\ \forall j \in V_2$. Since $(\bar{x}_1, \bar{x}_2)$ is in the relative interior of $\{(x_1,x_2) \in [0,1]^2 \mid \bar{a}_1 x_1 + \bar{a}_2 x_2 + \bar{c} = 0\}$, $(\bar{x}, \bar{y}, \bar{w})$ cannot be an extreme point of S. □

Proof of part (ii) of Theorem 3.1

First, we prove that the two-variable sets we encounter after fixing variables are SOCr.

Proposition 7. *Let* $S_0 = \{(x,y) \in [0,1]^2 \mid\ ax + by + qxy + c = 0\}$. *Then,* $\operatorname{conv}(S_0)$ *is SOCr.*

Proof. We may assume $S_0 \neq \emptyset$ and $q \neq 0$, otherwise the result follows trivially. Define $r = -b/q,\ s = -a/q$ and $\tau = (ab - cq)/q^2$ to write $ax + by + qxy + c = 0$ equivalently as

$$(x - r)(y - s) = \tau. \tag{3.5}$$

If $\tau = 0$, then (3.5) is equivalent to $x = r$ or $y = s$. In this case, $S_0 = \{(x,y) \in [0,1]^2 \mid x = r\} \cup \{(x,y) \in [0,1]^2 \mid y = s\}$ and hence $\operatorname{conv}(S_0)$ is a polytope. Suppose $\tau > 0$ (if $\tau < 0$, we multiply (3.5) by -1 and repeat the same proof with $x - r$ and τ replaced with $-(x - r)$ and $-\tau$). Either $x - r, y - s \geq 0$ or $x - r, y - s \leq 0$. Thus, $S_0 = S_0^> \cup S_0^<$, where $S_0^> = \{(x,y) \in [0,1]^2 \mid x - r, y - s \geq 0,\ (3.5)\}$ and $S_0^< = \{(x,y) \in [0,1]^2 \mid x - r, y - s \leq 0,\ (3.5)\}$. Next, we show that if $S_0^> \neq \emptyset$, then $\operatorname{conv}(S_0^>)$ is SOCr.

Using that $4uv = (u+v)^2 - (u-v)^2$, we can rewrite (3.5) as

$$\sqrt{[(x-r)-(y-s)]^2 + (2\sqrt{\tau})^2} = (x-r)+(y-s).$$

It now follows from Lemma 10 that $\mathrm{conv}(S_0^>) = \mathrm{conv}(S_1^>) \cap \mathrm{conv}(S_2^>)$, where

$$S_1^> = \{(x,y) \in [0,1]^2 \,|\, x-r, y-s \geq 0,\ \sqrt{[(x-r)-(y-s)]^2 + (2\sqrt{\tau})^2} \leq (x-r)+(y-s)\}$$
$$S_2^> = \{(x,y) \in [0,1]^2 \,|\, x-r, y-s \geq 0,\ \sqrt{[(x-r)-(y-s)]^2 + (2\sqrt{\tau})^2} \geq (x-r)+(y-s)\}.$$

Notice that $S_1^>$ is SOCr. Also, as the square root term in the definition of $S_2^>$ is a convex function in x and y, it follows from Lemma 11 that $S_2^>$ is a polytope. Thus, $\mathrm{conv}(S_0^>)$ is SOCr. Similarly, we can prove that $\mathrm{conv}(S_0^<)$ is SOCr by repeating the arguments above after replacing $x-r, y-s$ with $-(x-r), -(y-s)$. Therefore, $\mathrm{conv}(S_0) = \mathrm{conv}(S_0^> \cup S_0^<) = \mathrm{conv}(\mathrm{conv}(S_0^>) \cup \mathrm{conv}(S_0^<))$ is SOCr by Lemma 12. $\square$

Proposition 8. *Let* $S_0 = \{(x,y) \in [0,1]^2 \,|\ y = a_0 + a_1 x + a_2 x^2\}$. *Then* $\mathrm{conv}(S_0)$ *is SOCr.*

Proof. We may assume $S_0 \neq \emptyset$ and $a_2 \neq 0$, otherwise the result follows trivially. By completing squares, we can write $y = a_0 + a_1 x + a_2 x^2$ equivalently as $(x + 0.5a_1/a_2)^2 - (a_1/2a_2)^2 + a_0/a_2 = y/a_2$, and then as

$$(x+\bar{a})^2 = t \ \Leftrightarrow\ \sqrt{(x+\bar{a})^2 + \left(\frac{t-1}{2}\right)^2} = \frac{t+1}{2}, \tag{3.6}$$

where $\bar{a} = 0.5a_1/a_2,\ t = y/a_2 + (a_1/2a_2)^2 - a_0/a_2$, using that $4t = (t+1)^2 - (t-1)^2$. It now follows from Lemma 10 that $\mathrm{conv}(S_0) = \mathrm{conv}(S_1) \cap \mathrm{conv}(S_2)$, where

$$S_1 = \{(x,y) \in [0,1]^2 \,|\, \sqrt{(x+\bar{a})^2 + \left(\frac{t-1}{2}\right)^2} \leq \frac{t+1}{2}\}$$
$$S_2 = \{(x,y) \in [0,1]^2 \,|\, \sqrt{(x+\bar{a})^2 + \left(\frac{t-1}{2}\right)^2} \geq \frac{t+1}{2}\}.$$

Notice that S_1 is SOCr. Also, as the square root term in the definition of S_2 is a convex function in x and y (because t is an affine function of y), it follows from Lemma 11 that S_2 is a polytope. Thus, $\mathrm{conv}(S_0)$ is SOCr. $\square$

Proposition 9. *Let* $S_0 = \{(x, y, w) \in [0, 1]^3 \,|\, ax + by + qw + c = 0,\ w = xy\}$. *Then,* $\mathrm{conv}(S_0)$ *is SOCr.*

Proof. If $q \neq 0$, then we can write

$$S_0 = \{(x, y, w) \in [0, 1]^2 \times \mathbb{R} \,|\, (x, y) \in B_0,\ w = (-c - ax - by)/q\}, \tag{3.7}$$

where $B_0 = \{(x, y) \in [0, 1]^2 \,|\, ax + by + qxy + c = 0\}$. (Note that the bounds on w are automatically enforced in (3.7) and it is sufficient to say $w \in \mathbb{R}$). Hence, by Proposition 7 and Lemma 14, $\mathrm{conv}(S_0)$ is SOCr.

Now, suppose $q = 0$. Four cases: (i) $a, b = 0$. In this case, we may assume $c = 0$, otherwise $S_0 = \emptyset$. Then, $S_0 = \{(x, y, w) \in [0, 1]^3 \,|\, w = xy\}$, in which case $\mathrm{conv}(S_0)$ is a well known polytope given by the McCormick envelope. (ii) $a = 0$, $b \neq 0$. In this case, if $-c/b \notin [0, 1]$, then S_0 is infeasible. Otherwise, this case is trivial. (iii) $a \neq 0$, $b = 0$. Similar to previous case. (iv) $a \neq 0$ and $b \neq 0$. In this case, we can solve $ax + by + c = 0$ for x, i.e., $x = (-c - by)/a$. Let $[\alpha, \beta]$ be the bounds on y such that the line $ax + by + c = 0$ intersects the $[0, 1]^2$ box. If $\alpha = \beta$, then we can set $y = \alpha$ and the result follows trivially. Otherwise, substitute in $w = xy$ to rewrite S_0 as following

$$S_0 = \{(x, y, w) \in \mathbb{R} \times [\alpha, \beta] \times [0, 1] \,|\, (y, w) \in B_0,\ x = (-by - c)/a\},$$

where $B_0 = \{(y, w) \in [\alpha, \beta] \times [0, 1] \,|\, w = (-c/a)y - (b/a)y^2\}$. Now, it is straightforward via Proposition 8 (affinely scale y to have bound of $[0, 1]$) and Lemma 14 that $\mathrm{conv}(S_0)$ is a SOCr set. $\square$

Now we are ready to prove part (ii) of Theorem 3.1.

Proposition 10. *Let S be the set defined in (3.3). Then $\operatorname{conv}(S)$ is SOCr.*

Proof. By Proposition 6, we can fix various sets of x and y variables that corresponds to the U sets and prove that the convex hull of each of these sets is SOCr. Case (i): $|U| = 1$. In this case, the set of unfixed variables satisfy a set of linear equations. Thus this set is clearly SOCr. Case (ii): $U = \{(i_0, j_0)\}$, where $(i_0, j_0) \in E$. In this case, the set of unfixed variables satisfy the following constraints:

$$
\begin{aligned}
ax_{i_0} + by_{j_0} + qw_{i_0j_0} + c &= 0, && (3.8)\\
w_{i_0j_0} &= x_{i_0}y_{j_0} && (3.9)\\
w_{ij_0} &= \bar{x}_i y_{j_0} \ \forall (i, j_0) \in E, i \neq i_0 && (3.10)\\
w_{i_0j} &= \bar{y}_j x_{i_0} \ \forall (i_0, j) \in E, j \neq j_0, && (3.11)
\end{aligned}
$$

where the bound constraints on w_{ij_0} and w_{i_0j} variables are not needed explictly. Thus, by Proposition 9 and Lemma 14, the above set is SOCr. Thus, by Lemma 12, we obtain that $\operatorname{conv}(S)$ is SOCr. □

3.3.2 Proof of Theorem 3.2

In order to prove Theorem 3.2 it is sufficient to prove that:

$$\text{proj}_{x,y,w}\left(S^{SDP}\right) \supseteq S^{SOCP} \tag{3.12}$$

and

$$S^{QBP} \supseteq S^{SOCP}. \tag{3.13}$$

We prove these two containments next.

Proposition 11. *For any BBP, (3.12) holds.*

Proof. In order to prove (3.12), it is convenient to introduce some notation. Let H be the matrix variable representing $\begin{bmatrix} x \\ y \end{bmatrix} [x^\top y^\top]$. We write $w = \text{proj}_E(H)$, to imply that if $(i,j) \in E$, then $w_{ij} = \frac{1}{2}\left(H_{i(j+n_1)} + H_{(j+n_1)i}\right)$.

Then the standard SDP relaxation may be written as:

$$\begin{aligned} \sum_{ij \in E} (Q_k)_{ij} w_{ij} + a_k^\top x + b_k^\top y + c_k &= 0, \; k \in [m] && (3.14) \\ \text{proj}_E(H) &= w && (3.15) \\ \begin{bmatrix} H & [x^\top y^\top] \\ \begin{bmatrix} x \\ y \end{bmatrix} & 1 \end{bmatrix} &\succeq 0. && (3.16) \end{aligned}$$

Let

$$T^k := \{(x, y, H, w) \,|\, (3.14) \text{ corresponding to } k, (3.15), \text{ and } (3.16)\}$$

and as before let

$$S^k := \{(x, y, w) \,|\, (3.14) \text{ corresponding to } k, w_{ij} = x_i y_j \; \forall (ij) \in E\}.$$

Then by construction

$$\text{proj}_{x,y,w}\left(T^k\right) \supseteq \text{conv}(S^k). \tag{3.17}$$

Next we need the following:

Claim 1 $\bigcap_{k=1}^m \text{proj}_{x,y,w}\left(T^k\right) = \text{proj}_{x,y,w}\left(\bigcap_{k=1}^m \left(T^k\right)\right)$: Trivially we have that,

$$\bigcap_{k=1}^m \text{proj}_{x,y,w}\left(T^k\right) \supseteq \text{proj}_{x,y,w}\left(\bigcap_{k=1}^m \left(T^k\right)\right),$$

holds.

We now verify the converse. For some $(\bar{x}, \bar{y}, \bar{w}) \in \text{proj}_{x,y,w}\left(T^k\right)$, let

$$\mathcal{H}^k(\bar{x}, \bar{y}, \bar{w}) := \left\{H \mid (\bar{x}, \bar{y}, \bar{w}, H) \in T^k\right\}.$$

Then observe that $\mathcal{H}^k(\bar{x}, \bar{y}, \bar{w})$ is the set of matrices H satisfying

$$\text{proj}_E(H) = \bar{w} \tag{3.18}$$

$$\begin{bmatrix} H & [\bar{x}^\top \bar{y}^\top] \\ \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} & 1 \end{bmatrix} \succeq 0. \tag{3.19}$$

Thus $\mathcal{H}^k(\bar{x}, \bar{y}, \bar{w})$ is independent of k, i.e., if $(\bar{x}, \bar{y}, \bar{w}) \in \bigcap_{k=1}^m \text{proj}_{x,y,w}\left(T^k\right)$ then $\mathcal{H}^{k_1}(\bar{x}, \bar{y}, \bar{w}) = \mathcal{H}^{k_2}(\bar{x}, \bar{y}, \bar{w})$ for all $k_1 \neq k_2$. Therefore in particular, if $(\bar{x}, \bar{y}, \bar{w}) \in \bigcap_{k=1}^m \text{proj}_{x,y,w}\left(T^k\right)$, then there exists $\bar{H}$ such that $(\bar{x}, \bar{y}, \bar{w}, \bar{H}) \in \bigcap_{k=1}^m T^k$. Thus, $(\bar{x}, \bar{y}, \bar{w}) \in \text{proj}_{x,y,w}\left(\bigcap_{k=1}^m T^k\right)$. $\diamond$

Now, we return to the proof of the original statement. Intersecting (3.17) for all $k \in [m]$ we obtain,

$$\text{proj}_{x,y,w}\left(S^{SDP}\right) = \text{proj}_{x,y,w}\left(\bigcap_{k=1}^m \left(T^k\right)\right) = \bigcap_{k=1}^m \text{proj}_{x,y,w}\left(T^k\right) \supseteq \bigcap_{k=1}^m \text{conv}(S^k) = S^{SOCP},$$

where the first equality is by definition of S^{SDP}, the second equality via Claim 1, the inequality is due to (3.17) and the last equality is by definition of S^{SOCP}. $\square$

Proposition 12. *For any BBP, (3.13) holds.*

Proof. Recall that S^{QBP} is the set

$$\left\{(x, y, w) \in [0,1]^{n_1+n_2+|E|} \mid \sum_{(ij)\in E} (Q_k)_{ij} w_{ij} + a_k^\top x + b_k^\top y + c_k = 0 \; k \in [m]\right\} \tag{3.20}$$

$$\bigcap \text{conv}\left(\{(x, y, w) \in [0,1]^{n_1+n_2+|E|} \mid w_{ij} = x_i y_j \; \forall (i,j) \in E\}\right). \tag{3.21}$$

Let

$$T^k := \{(x, y, w) \in [0, 1]^{n_1+n_2+|E|} \mid \text{(3.20) corresponding to } k, \text{(3.21)}\}$$

and let

$$S^k := \{(x, y, w) \mid \text{(3.14) corresponding to } k, w_{ij} = x_i y_j \; \forall (ij) \in E\}.$$

Then by construction

$$T^k \supseteq \text{conv}(S^k). \tag{3.22}$$

Intersecting (3.22) for all $k \in [m]$ we obtain,

$$S^{QBP} = \bigcap_{k=1}^{m} T^k \supseteq \bigcap_{k=1}^{m} \text{conv}(S^k) = S^{SOCP}.$$

□

3.4 Proposed branch-and-bound algorithm

In this section, we discuss some details of our proposed branch-and-bound algorithm to solve BBP (3.1).

3.4.1 Node selection and partitioning strategies

The most common node selection rule used in the literature is the so-called *best-bound-first*, in which a node with the least lower bound (assuming minimization) is chosen for branching. Other rules may include selection of nodes that have the potential of identifying good feasible solutions earlier. In our computational experiments, we only use best-bound-first rule. Also, we use the most simple partitioning operation: rectangular. Example of other operation adopted in the literature are conical and simplicial [89].

3.4.2 Variable selection and point of partitioning

A simple rule for variable selection is to choose a variable with largest range. Another common rule is to prioritize the variable that is most responsive for the approximation error of nonlinear terms. For example, suppose we are optimizing in the extended space of (x, y, w), then we could chose x_i (or y_j) for which the absolute error $|\bar{w}_{ij} - \bar{x}_i\bar{y}_j|$ is maximized over the set of all possible pairs (i, j), where $(\bar{x}, \bar{y}, \bar{w})$ is the relaxation solution for the current node. We refer to this rule as the *gap-error-rule*.

Once the variable is selected, say x_1 (without loss of generality), we can list three standard rules for choosing the partitioning point:

Bisection: partition at the mid point of the domain of x_1 in the current node.

Maximum-deviation: partition at $\bar{x}_1$, where $(\bar{x}, \bar{y}, \bar{w})$ is the relaxation solution for the current node.

Incumbent: partition at x_1^*, where (x^*, y^*, w^*) is the current best feasible solution, if x_1^* is in the range of x_1 in the current node.

Combination of the above rules have also been proposed. For example, Tawarmalani et al. [122] propose a rule that is a convex combination of bisection and maximum-deviation branching rules (biased towards the maximum-deviation), and uses incumbent branching whenever possible.

In our proposed algorithm, we use specialized variable and branching point selection rules, which use information collected from multiple disjunctions and, therefore, take into account the coefficients of the constraints in the model in addition to the variable ranges at the current node.

New proposed rule Note that we always branch on only one set of variables, either x or y. We describe our rule assuming we are branching on the x variables. To further ease exposition, we explain our proposed branching rules for the root node, i.e., we assume that all variables range from 0 to 1. Consider the three-variable set:

$$S_0 = \{(x_1, y_1, w_{11}) \in \mathbb{R}^3 \,|\, qw_{11} + ax_1 + by_1 + c = 0,\ w_{11} = x_1 y_1\},$$

which is obtained by fixing x_i, y_j to either 0 or 1 in (EQ), $\forall i \in V_1 \setminus \{1\},\ \forall j \in V_2 \setminus \{1\}$. Like the proof of Proposition 9, there are two cases of interest.

- $q \neq 0$. In this case, w_{11} can be written as affine function of x_1 and y_1. We can then write the projection of S_0 in the space of (x_1, y_1) as (we drop the indices to simplify notation, we also drop the word 'Proj')

  $$S_0 = \{(x, y) \in [0,1]^2 \,|\ (x - r)(y - s) = \tau\},$$

  where r, s, τ are constants. The equation $(x - r)(y - s) = \tau$ represents a hyperbola with asymptotes $x = r$ and $y = s$. Two typical instances are plotted in Figure 3.3-3.4, where the continuous thick portion of the curves represents S_0 and the whole dotted areas represent $\mathrm{conv}(S_0)$. Our goal is to branch at a point that maximizes the eliminated area upon branching.

  Case 1: Both branches of a hyperbola intersect with the $[0,1]^2$ box. Let x_l (resp. x_u) be the x-coordinate of the intersection point of the left (resp. right) branch with either of the lines $y = 0$ or $y = 1$. The plot on Figure 3.3 suggests that branching x at any point $x_0 \in [x_l, x_u]$ is a reasonable choice for the case where both branches of the hyperbola intersect the $[0,1]^2$ box. Indeed, such branching would eliminate the entire dotted area between the two branches of the curve.

  Case 2: One branch of hyperbola intersects with the $[0,1]^2$ box. For the case where only one branch intersects the $[0,1]^2$ box, as illustrated in Figure 3.4, we could in principle compute C that maximizes the area of the triangle $\triangle_{ABC}$. To simplify the rule and avoid excessive computations, we simply choose C to be the point at

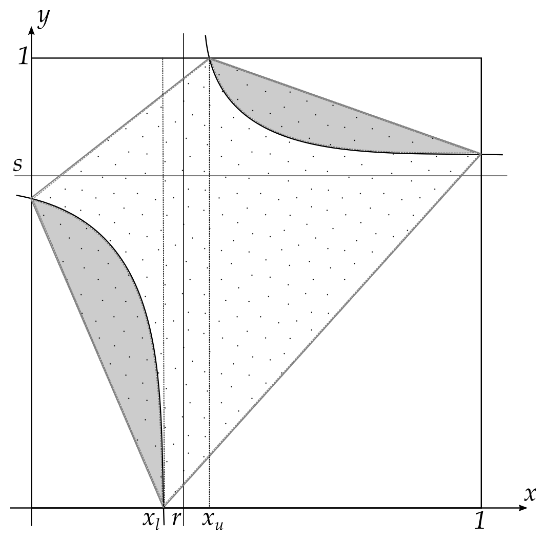

Figure 3.3: Convex hull of the set defined by the intersection of two branches of a hyperbola with the $[0,1]^2$ box. Here, x_l (resp. x_u) is the x-coordinate of the intersection point of the left (resp. right) branch with the line $y = 0$ (resp. $y = 1$).

which the tangent line to the curve is parallel to the line AB. Moreover, for points in some interval $[x_l, x_u]$ containing x_c, the area of the triangle $\triangle_{ABC}$ does not change much, implying that every point in $[x_l, x_u]$ may be a good choice to branch at. In our computational experiments, we compute x_l and x_u such that $x_c - x_l = \gamma(x_c - x_a)$ and $x_u - x_c = \gamma(x_b - x_c)$ with $\gamma = 2/3$.

- $q = 0$ and $a \neq 0$ or $b \neq 0$. Without loss of generality assume $b \neq 0$. In this case, y_1 is an affine function of x_1 as shown in proof of Proposition 9. Thus, we can study S_0 in the space of (x_1, w_{11}), where it is defined by a parabola and we adopt the same rule defined for the case of Figure 3.4, i.e., choose points x_l and x_u as a function of x_a and x_b. If the parabola intersects the $[0,1]^2$ box in more than two points, we define A and B to be the left and right most intersection points.

  Note that if $a \neq 0$, then x_1 is an affine function or y_1. We can identify appropriate points in the y_1 space as above and then translate them to the x space via the affine function.

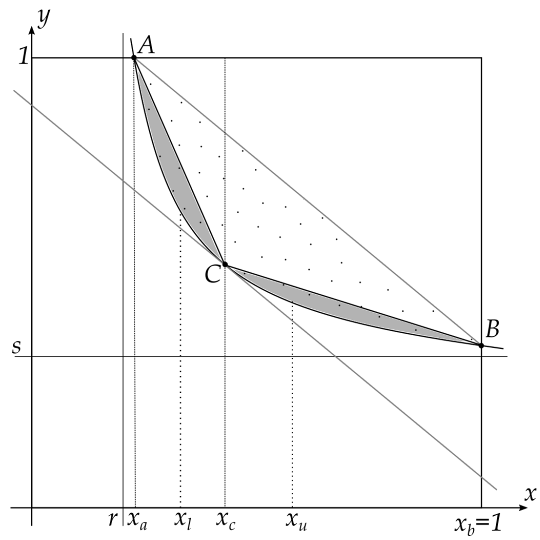

Figure 3.4: Convex hull of the set defined by the intersection of a single branch of a hyperbola with the $[0,1]^2$ box. Let A and B are the intersection points of the curve with the $[0,1]^2$ box and C is the point of the curve at which the tangent line is parallel to AB. Then, x_a, x_b and x_c are the projections of A, B and C onto the x axis.

Thus, corresponding to every three-variable set S_0, we associate (i) an x-variable x_i, (ii) an interval $[x_l, x_u]$ within the domain of x_i and (iii) we also approximately compute the area of conv(S_0), (either in the space of (x_1, y_1), if $q \neq 0$, or in the space of (x_1, w_{11}), if $q = 0$), referred to as A_0. The actual area we use is that of the polyhedral outer approximation as will be discussed in Section 3.5.2.

Once the above data is collected for all disjunctions, we use the following Algorithm to decide on the variable to branch on and the point of partitioning for this variable.

Algorithm 1 Branching rule

Input: $\delta = 1/K$, for some positive integer K. Let $\varepsilon_1, \varepsilon_2 > 0$.
Let $A_{ik} = 0,\ i \in [n_1],\ k \in [K]$. Let $p_i = 0,\ i \in [n_1]$
Define $I_{ik} = [(k-1)\delta, k\delta]$, for $k \in [K]$ (which defines a partition of the range of x_i).
for Each disjunctions S_0 **do**
    Compute (a) the index i of x-variable corresponding to S_0, (b) domain $[x_l, x_u]$ and (c) the area A_0.
    Set $p_i = p_i + 1$
    If $[x_l, x_u] \cap I_{ik} \neq \emptyset$ for some $k \in [K]$, set $A_{ik} = A_{ik} + A_0$.
end for

for $i \in [n_1]$ **do**
    if $\frac{p_i}{\sum_{l=1}^{n_1} p_l} < \varepsilon_1$ **then**
        variable i is declared irrelevant.
    end if
end for
Let $(i^*, k^*) \in \mathrm{Argmax}\{A_{i,k} \,|\, i \in [n_1], i \text{ is not irrelevant}, k \in [K]\}$
if $A_{i^*k^*} \geq \varepsilon_2$ **then**
    Branch on the variable x_{i^*} at the mid point of the interval $I_{i^*k^*}$.
else
    Use the bisection rule.
end if

In our computational experiments, whenever we use Algorithm 1, we set $\varepsilon_1 = 0.01,\ \varepsilon_2 = 1/16$ and $K = 8$. Our implementation is naive, and we have not tried to fine tune any of these parameters.

3.5 Computational experiments

3.5.1 Finite Element Updating Model

The instances of BBP that we use come from finite element (FE) model updating in structural engineering. The goal is to update the parameter values in an FE model, so that the model provides same resonance frequencies and mode shapes that are physically measured from vibration testing at the as-built structure. In this study we adopt the modal dynamic residual formulation, for which the details can be found in [141]. The formulation is briefly summarized as follows.

Consider the model updating of a structure with m number of degrees-of-freedom (DOFs). Corresponding to stiffness parameters that are being updated, the (scaled) updating variables are first denoted as $x \in [-1,1]^{n_1}$. Since only some DOFs can be instrumented, we suppose n_2 of those are not instrumented, leaving $m - n_2$ of them as instrumented. In the meantime, it's assumed that n_3 number of vibration modes are measured/observed from the vibration testing data. For each l-th measured mode, $\forall l \in \{1, \dots, n_3\}$, the experimental results provide λ_l as the square of the (angular) resonance frequency, and $\bar{y}^l \in \mathbb{R}^{m-n_2}$ as the mode shape entries at the instrumented DOFs. In mathematical terms, the modal dynamic residual formulation can be stated as the problem of simultaneously solving the following set of equations on stiffness updating variables $x \in [-1,1]^{n_1}$ and (scaled) unmeasured mode shape entries $y^l \in [-2,2]^{n_2}$, $\forall l \in [n_3]$:

$$[K_0 + \sum_{i=1}^{n_1} x_i K_i - \lambda_l M] \begin{bmatrix} \bar{y}^l \\ y^l \end{bmatrix} = 0,\ l \in [n_3], \tag{3.23}$$

where $M, K_0, K_i \in \mathbb{R}^{m \times m}$, $\forall i \in [n_1]$, $\lambda_l \in \mathbb{R}_+$ and $\bar{y}^l \in \mathbb{R}^{m-n_2}$, $\forall l \in [n_3]$, are problem data. In practice, (3.23) is unlikely to have a feasible solution set of x and y^l, $l \in [n_3]$, because of modeling and measurement inaccuracies. Therefore, we convert the problem of solving (3.23) into an optimization problem that aims to minimize the sum of the residuals, i.e., the absolute difference between left and right-hand-side of each equation. After some affine transformations and simplifications, this optimization problem can be stated as following:

$$\begin{aligned}\min \quad & \sum_{k=1}^{m} z_k \\ \text{s.t.} \quad & |x^\top Q_k y + a_k^\top x + b_k^\top y + c_k| = z_k,\ k \in [m] \\ & x \in [0,1]^{n_1}, y \in [0,1]^{n_2},\end{aligned} \tag{3.24}$$

where n_2 and m correspond to n_2n_3 and mn_3, respectively, in the notation of (3.23). Finally, (3.24) is equivalent to the following BBP.

$$\begin{aligned}\min \quad & \sum_{k=1}^{m} z'_k + z''_k \\ \text{s.t.} \quad & x^\top Q_k y + a_k^\top x + b_k^\top y + c_k = z'_k - z''_k,\ k \in [m] \\ & x \in [0,1]^{n_1}, y \in [0,1]^{n_2}. \\ & 0 \le z'_k, z''_k \le u,\ k \in [m].\end{aligned} \tag{3.25}$$

Instances:

The simulated structural example is similar to the planar truss structure in [141]. In order to simulate measurement noise, we add a normal-distributed random variable to the parameters λ^l and $\bar{y}^l$, $\forall l \in [n_3]$, with mean zero and variance equal 2% of its actual value. In our case there are six modes, i.e., $n_3 = 6$. By taking different values for n_2, we then generate ten instances whose number of variables and constraints are given in Table 3.1.

3.5.2 Simplifying S^{SOCP}

A lighter version of S^{SOCP}

According to Remark 3, the number of disjunction needed to model the convex hull of a single bilinear equation can be computationally prohibitive for many instances of interest.

Table 3.1: Instances description

Inst	# of x-variables	# of y-variables	# of equations	# of bilinear terms
inst1	6	180	312	990
inst2	6	180	312	954
inst3	6	168	312	966
inst4	6	168	312	972
inst5	6	156	312	900
inst6	6	144	312	780
inst7	6	132	312	756
inst8	6	132	312	756
inst9	6	120	312	684
inst10	6	120	312	684

To overcome this issue, in our computational experiments, we write the convex hull of each row only in the space of the variable appearing in it. In particular, for constraint k we work with $G(V^k, E^k)$, where V^k is the set of variables appearing in constraint k and E^k represent the complete bipartite graph between the x and y variables appearing in V^k. This possibly weaker relaxation is much more computationally cheaper that S^{SOCP} for our instances due to the sparsity on the coefficients of each bilinear equation. We denote this relaxation as light $- S^{SOCP}$.

Polyhedral outer approximation

As shown in Proposition 7 and Proposition 8, all the sets obtained after fixings are SOCr. Some are polyhedral while many of the others are not. Since linear programming techniques are more efficient and robust, than the non-linear counterpart, we outer approximate the non-polyhedral sets by polyhedral sets.

As shown in proof of Proposition 10, all the non-linear sets that we need to convexify in order to obtain the convex hull of the set S defined in (EQ) are of the form

$$\begin{aligned} S_{i_0j_0} = \{(x,y,w) \in [0,1]^{n_1+n_2+n_1n_2} \,|\, x_i, y_j \in \{0,1\},\ \forall i \in V_1 \setminus \{i_0\},\ \forall j \in V_2 \setminus \{j_0\}, \\ \bar{q}w_{i_0j_0} + \bar{a}x_{i_0} + \bar{b}y_{i_0} + \bar{c} = 0,\ w_{ij} = x_iy_j,\ i \in V_1,\ j \in V_2\}, \end{aligned}$$

for some $(i_0, j_0) \in E$. Without loss of generality, suppose $i_0 = 1$ and $j_0 = 1$, in which case we want to outer approximate the following set $S_0 = \{(x_1, y_1, w_{11}) \in \mathbb{R}^3 \,|\, qw_1 + ax_1 + by_1 + c = 0,\ w_{11} = x_1y_1\}$. There are two cases of interest. The first case occurs when $q \neq 0$. In this case, w_{11} is an affine functions of x_1 and y_1 as following: $w_{11} = (-c - ax_1 - by_1)/q$; $w_{1j} = x_1y_j,\ \forall j \in [n_2]$; $w_{i1} = x_iy_1,\ \forall i \in [n_1]$; and $w_{ij} = x_iy_j,\ \forall i \in [n_1] \setminus \{1\},\ \forall j \in [n_2] \setminus \{1\}$. Hence, we only need to approximate $\mathrm{conv}(S_0)$ in the space of (x_1, y_1). If both branches of the hyperbola defined by $qx_1y_1 + ax_1 + by_1 + c = 0$ intersect the $[0, 1]^2$ box, than $\mathrm{conv}(S_0)$ is polyhedral. Suppose only one branch of the hyperbola intersects the box. Then, we outer approximate $\mathrm{conv}(S_0)$ by using tangent lines to the curve. In our implementation, we only use the tangent lines at the intersection points of the curve with the box, see Figure 3.5. More tangent lines could be added to better approximate $\mathrm{conv}(S_0)$, but based on our preliminary experience on our instances it does not make significant difference.

The second case of interest is $q = 0$ and $a \neq 0$ (or $b \neq 0$) for which we can rewrite S_0 as $S_0 = \{(x_1, y_1, w_{11}) \in [0, 1]^3 \,|\, aw_{11} = -by_1^2 - cy_1,\ ax_1 = -by_1 - c\}$. In this case, x_1 is an affine function of y_1 and we only need to approximate $\mathrm{conv}(S_0)$ in the space of (y_1, w_{11}), where $aw_{11} = -cy_1 - by_1^2$ defines a parabola as shown in Figure 3.6. As in the previous case, we outer approximate the curve by using tangent lines to the curve as illustrated in Figure 3.6.

3.5.3 Computation results

Software and Hardware

All of our experiments were ran on a Windows 10 machine with 64-bit operating system, x64 based processor with 2.19GHz, and 32GB RAM. We call MOSEK via CVX from MATLAB R2015b to solve SDPs. We used Gurobi 7.5.1 to solve LPs and integer programs. We used BARON 15.6.5 (with CPLEX 12.6 as LP solver and IPOPT as nonlinear solver) as our choice of commercial global solver, which we call from MATLAB R2015b.

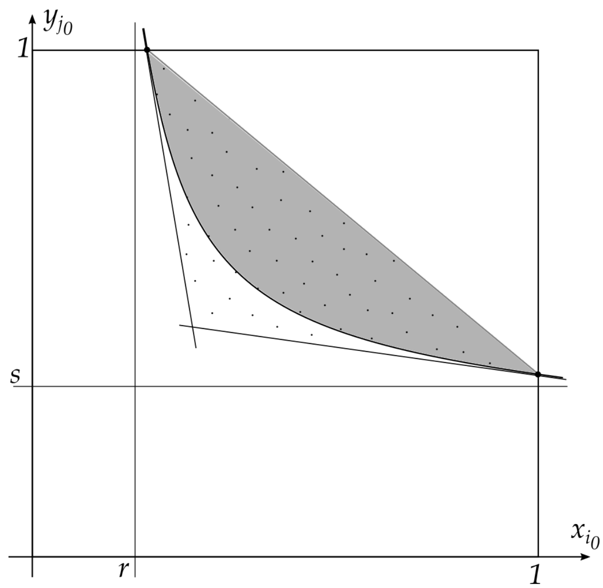

Figure 3.5: Convex hull of the set defined by the intersection of one branch of a hyperbola with the $[0, 1]^2$ box, and its tangential linear outer approximation.

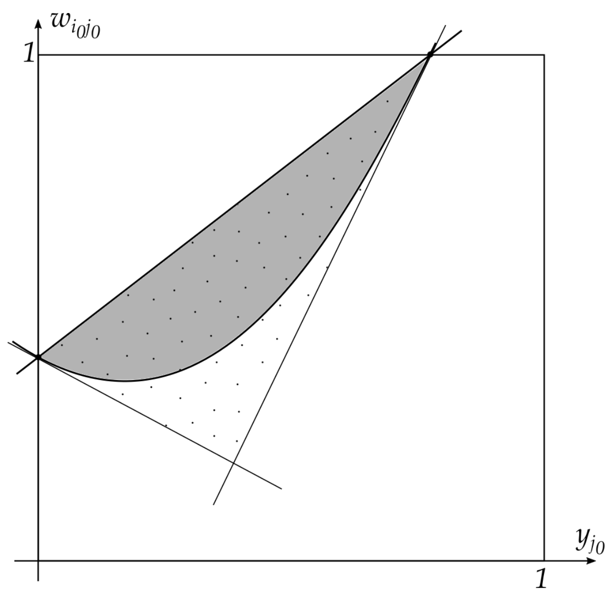

Figure 3.6: Convex hull of the set defined by the intersection of parabola with the $[0, 1]^2$ box, and its tangential linear outer approximation.

Root node

We assess the strength of our proposed polyhedral outer approximation of light $-\ S^{SOCP}$ relaxation (defined in Section 3.5.2 and referred as SOCP in the tables) against the classical SDP and McCormick (Mc) relaxations. The numerical results are reported in Table 3.2, where SDP+Mc denotes the the intersection of SDP and Mc relaxations. Similarly, SOCP+Mc denotes the intersection of SOCP and Mc relaxations (since we are not using S^{SOCP}, this could potentially be stronger than SOCP).

Table 3.2: Root relaxations

	Mc		SDP		SDP+Mc		SOCP		SOCP+Mc	
Inst	Bound	Time	Bound	Time	Bound	Time	Bound	Time	Bound	Time
1	0.17771	0.07	0.17771	1.81	0.17771	35.89	0.17793	17.59	0.17793	18.42
2	0.00000	0.05	0.00000	1.70	0.00000	38.98	0.00000	20.93	0.00000	21.14
3	0.27543	0.07	0.27194	1.81	0.27543	44.02	0.28202	16.22	0.28202	49.61
4	0.10095	0.08	0.10012	2.14	0.10095	36.13	0.10101	20.71	0.10101	25.87
5	0.34766	0.05	0.34766	1.67	0.34766	31.58	0.34925	13.17	0.34925	12.88
6	0.97758	0.05	0.91629	1.80	0.97758	28.47	1.00267	11.64	1.00267	11.07
7	1.73437	0.07	1.70329	1.38	1.73437	25.29	1.74015	10.76	1.74015	11.68
8	1.99887	0.07	1.97107	1.30	1.99887	21.95	2.01260	17.53	2.01260	21.51
9	1.89400	0.05	1.89222	1.17	1.89400	22.94	1.90191	10.53	1.90191	9.32
10	2.41036	0.05	2.40658	1.16	2.41036	18.95	2.41959	10.07	2.41959	12.29

As we see, SOCP produces the best dual bounds among SDP, Mc and SDP+Mc. Also, SOCP runs faster than SDP+Mc for all the instances. Finally, SOCP+Mc produces no better bounds than SOCP alone.

A strong relaxation can be obtained by partitioning the domain of some variables and writing a MILP formulation to model the union of McCormick relaxations over each piece [98, 51]. We call it McCormick Discretization and use the MILP formulation with binary expansion. We only partition the domain of variables x_i's as the number of x variables is much smaller than the number of y variables for all of our instances. In Table 3.3, T defines the level of discretization, meaning that the range of each variable x_i is partitioned into $2^T + 1$ uniform sub-intervals. This relaxation becomes tighter as T increases. However, the MILP that need to be solved becomes harder since the number of binary variables increases as a function of T. Thus, we give Gurobi a time limit of 10 hours, which is the amount of time given to all the branch-and-bound algorithm that we report in Section 3.5.3 below. Table 3.3 reports the computational results, where the asterisk signalizes that Gurobi reached the time limit with the given level of discretization. If this is the case, then we report the MILP dual bound reported by the solver, which is a valid dual bound for our problem. The last column displays the best bound obtained among all the levels of discretizations reported.

Table 3.3: McCormick discretization: dual bounds

Inst	T=6	T=8	T=10	T=12*	T=14*	T=16*	Best
1	0.18611	0.20512	1.11852	1.85387	1.40586	0.96121	1.85387
2	0.00000	0.03133	1.05662	2.14709	1.38374	0.04654	2.14709
3	0.29443	0.33575	1.39375	2.14270	1.42642	1.42007	2.14270
4	0.10524	0.11387	1.21446	2.44853	1.63495	1.27218	2.44853
5	0.36159	0.47559	2.15416	3.40272	3.22915	2.67721	3.40272
6	1.25052	2.61325	4.16459	4.06782	3.96512	3.78165	4.16459
7	1.96682	2.17988	3.60737	4.92133	4.69632	4.47471	4.92133
8	2.48886	2.69510	3.63400	4.81890	4.48014	4.19095	4.81890
9	2.05584	2.42150	4.16064	5.54076	5.63110	5.15290	5.63110
10	2.57751	2.80795	4.07475	5.40977	5.28173	5.16376	5.40977

Clearly, McCormick discretization produces better results than $SOCP$. Therefore, if one does not want to use branch and bound, then McCormick discretization is the best option. However, as we see in the next section, better dual bounds can be obtained by combining SOCP with the new proposed branch-and-bound algorithm.

Branch-and-bound

We assess and compare the performance of the following methods:

- *BB*: This stands for our implementation of a branch-and-bound algorithm coded in Python. We use Gurobi as LP solver and run IPOPT at each node to search for feasible solutions. Our algorithm uses best-bound-first as node selection and rectangular partitioning. We consider three variants that differ from each other based on the relaxation adopted in each node and in the way variables and branching points are selected:

  - *SOCP-1*: Uses the polyhedral relaxation described in Section 3.5.2 with variable selection and the branching point given by Algorithm 1.

  - *SOCP-2*: Uses the same relaxation of BB-SOCP-1 above. The branching variable is selected according to the gap-error-rule explained in Section 3.4.2. Then uses the incumbent-rule for branching point selection, whenever possible, otherwise uses the maximum-deviation-rule.

- *SOCP-3*: Same as BB-SOCP-2 except that uses bisection for branching point selection.

- *BB-Mc*: Uses McCormick relaxation with gap-error-rule as branching variable selection rule and bisection for branching point selection.

The dual bounds from our computational experiments are reported in Table 3.4. The stopping criteria for all the methods was a time limit of 10 hours.

Table 3.4: Branch-and-bound methods: dual bounds

Inst	BB-SOCP-1	BB-SOCP-2	BB-SOCP-3	BB-Mc
1	2.50744	0.18473	0.18228	0.18343
2	2.86438	0.00000	0.00000	0.00000
3	3.13078	0.29109	0.28983	0.28884
4	3.11154	0.10526	0.10246	0.10410
5	3.78958	0.35253	0.35392	0.35405
6	4.63992	1.11105	1.09537	1.15191
7	5.26603	1.99569	1.88331	1.94949
8	5.13128	2.18546	2.18193	2.28761
9	6.10860	2.17509	2.08068	2.10144
10	5.77051	2.48039	2.45158	2.47965

The best dual bound for each instance is clearly given by BB-SOCP-1, which uses our proposed relaxation and branching rule. All the standard branching rules yield significantly worse bounds.

McCormick relaxation with BB-SOCP-1 branching rules

The computational results from Section 3.5.3, suggest that the good performance of BB-SOCP-1 is highly dependent on its branching rules, defined according to Algorithm 1. In this section we show that the branching rules of Algorithm 1 on them own are not enough to produce good dual bounds.

Consider the variant of BB-SOCP-1, reffered as BB-SOCP-Mc, which uses only McCormick relaxation and the same branching rule given by Algorithm 1. Thus, at each node, we collect data from each disjunction S_0, run Algorithm 1 to select the branching variable and the branching point, but we only use the McCormick inequalities to define the relaxation.

In Table 3.5, we compare the performance of BB-SOCP-1 and BB-SOCP-Mc. It becomes clear that the strength of BB-SOCP-1 does not come only from the branching rules of Algorithm 1 but also from our proposed relaxation. The discrepancy in the performance of BB-SOCP-1 and BB-SOCP-Mc means that, as the algorithm goes down the tree, the SOCP relaxation becomes much tighter than the McCormick relaxation.

Table 3.5: BB-SOCP-1 vs. McCormick relaxation with BB-SOCP-1 branching rules

	BB-SOCP-1		BB-SOCP-Mc	
Inst	Dual Bound	Gap (%)	Dual Bound	Gap (%)
1	2.50744	27.9	0.19776	94.3
2	2.86438	18.2	0.02752	99.2
3	3.13078	14.9	0.30514	91.7
4	3.11154	17.1	0.11188	97.0
5	3.78958	8.3	0.40497	90.2
6	4.63992	18.0	1.52070	73.1
7	5.26603	6.0	2.26765	59.5
8	5.13128	9.5	2.68861	52.6
9	6.10860	1.5	2.51461	59.5
10	5.77051	7.9	2.85232	54.2

Comparison of primal bounds and duality gaps

Finally, we report in Table 3.6 a summary of the performance of BB-SOCP-1, McCormick Discretization, BARON and BB-Mc. Recall that the stopping criteria for all the methods was a time limit of 10 hours. Also recall that primal solutions for BB-SOCP-1 and BB-Mc are obtained using IPOPT.

The primal bounds from all the three branch-and-bound methods are similar, suggesting that the solutions found are close to a global optimal. On the other hand, the dual bounds

Table 3.6: Primal bounds and duality gaps

	BB-SOCP-1			Mc Disc		BARON			BB-Mc		
Inst	Dual	Primal	Gap(%)	Dual	Gap(%)	Dual	Primal	Gap(%)	Dual	Primal	Gap(%)
1	2.50744	3.47847	27.9	1.85387	46.7	0.33122	3.47887	90.5	0.18343	3.47849	94.7
2	2.86438	3.49983	18.2	2.14709	38.6	0.52447	3.49931	85.0	0.00000	3.49983	100.0
3	3.13078	3.68103	14.9	2.14270	41.8	0.47599	3.68306	87.1	0.28884	3.73308	92.3
4	3.11154	3.75223	17.1	2.44853	34.7	0.78630	3.75297	79.0	0.10410	3.75225	97.2
5	3.78958	4.13277	8.3	3.40272	17.7	0.38396	4.13541	90.7	0.35405	4.28165	91.7
6	4.63992	5.66096	18.0	4.16459	26.4	2.26566	5.66053	60.0	1.15191	5.66096	79.7
7	5.26603	5.60009	6.0	4.92133	12.1	3.07096	5.60020	45.2	1.94949	5.69318	65.8
8	5.13128	5.67022	9.5	4.81890	15.0	2.70237	5.67025	52.3	2.28761	5.67252	59.7
9	6.10860	6.20343	1.5	5.63110	9.2	3.67301	6.20346	40.8	2.10144	6.29365	66.6
10	5.77051	6.26853	7.9	5.40977	13.1	2.94060	6.22639	52.8	2.47965	6.30477	60.7

from BB-SOCP-1 are significantly better than the dual bounds from all the other methods, which can be seem by comparing the duality gaps. In particular, the duality gap from BB-SOCP-1 is considerably smaller than the duality gap from Mc Disc, even though we are reporting the best dual bound obtained among all the levels of discretizations $T = 6, 8, \cdots, 16$, and the primal bound we use to compute the duality gap of Mc Disc is the best primal bound from BB-SOCP-1, BARON and BB-Mc. The standard branching, i.e., the McCormick relaxation with bisection, yields the worse performance for all the instances.

CHAPTER 4

THE CONVEX HULL OF A QUADRATIC CONSTRAINT OVER A POLYTOPE

The work presented in this chapter has already been submitted. See first version of the paper at: `http://www.optimization-online.org/DB_HTML/2018/12/7004.html`.

4.1 Introduction

A quadratically constrained quadratic program (QCQP) is an optimization problem in which the objective function is a quadratic function and the feasible region is defined by quadratic constraints. A variety of complex systems can be cast as an instance of a QCQP. Combinatorial problems like MAXCUT [62], engineering problems such as signal processing [61, 77], chemical process [71, 98, 5, 51, 67, 136] and power engineering problems such as the optimal power flow [31, 87, 41, 83] are just a few examples.

Solving non-convex QCQP to global optimality is a well-know NP-hard problem and a traditional approach is to use spacial branch-and-bound tree based algorithm. The computational success of any branch-and-bound tree based algorithm depends on the convexification scheme used at each node of the tree. Not surprisingly, there has been a lot of research on deriving strong convex relaxations for general-purpose QCQPs. The most common relaxations found in the literature are based on Linear programming (LP), second-order cone programing (SOCP) or semidefinite programming (SDP). Reformulation-linearization technique (RLT) [127, 128] is a LP-based hierarchy, Lasserre hierarchy or the sum-of-square hierarchy [86] is a SDP-based hierarchy which exactly solves QCQPs under some minor technical conditions and, recently, new LP and SOCP-based alternatives to sum of squares optimization have also been proposed [3]. While SDP relaxations are know to be strong, they don't always scale very well computationally. SOCP relaxations tend to be more computationally attractive, although they are often derived by further relaxing SDP relaxations [35].

Another direction of research focuses on convexification of functions, with the McCormick relaxation [95] being perhaps the most classic example. In this case, a constraint of the form $f(x) = b$ is replaced with $\breve{f}(x) \leq b$ and $\hat{f}(x) \geq b$, where $\breve{f}$ is a convex lower approximation and $\hat{f}$ is a concave upper approximation of f. While there have been a lot of work in function convexification (see for instance [4, 125, 120, 88, 20, 97, 9, 18, 14, 100, 49, 126, 118, 99, 136, 137, 90, 33, 44, 1, 68, 131, 8]) it is well-known that it does not necessarily yield the convex hull of the set $\{x \mid f(x) = b\}$. To the best of our knowledge, there have been much less work on explicit convexification of sets: [135, 110, 111, 134, 66, 82, 115, 48, 87, 34].

A related question when studying convex relaxations is that of representability of the exact convex hull of the feasible set: Is it LP, SOCP or SDP representable? In [55], we prove that the convex hull of the so-called bipartite bilinear constraint (which is a special case of a quadratic constraint) intersected with a box constraint is SOCP representable (SOCr). The proof yields a procedure to compute this convex hull exactly. Encouraging computational results are also reported in [55] in terms of obtaining dual bounds using this construction, which significantly outperform SDP and McCormick relaxations and also bounds produced by commercial solvers.

4.2 Our result

For an integer $n \geq 1$, we use $[n]$ to describe the set $\{1, \ldots, n\}$. For a set $G \subseteq \mathbb{R}^n$, we use conv(G), extr(G) to denote the convex hull of G and the set of extreme points of G respectively.

In this chapter, we generalize one of the main result in [55]. Specifically, we show that the convex hull of a *general* quadratic equation intersected with *any* bounded polyhedron is SOCr. Moreover the proof is constructive, therefore adding to the literature on explicit convexification in the context of QCQPs. The formal result is as following:

Theorem 4.1. *Let*

$$S := \{x \in \mathbb{R}^n \mid x^\top Q x + \alpha^\top x = g,\ x \in P\}, \tag{4.1}$$

where $Q \in \mathbb{R}^{n\times n}$ *is a symmetric matrix,* $\alpha \in \mathbb{R}^n$, $g \in \mathbb{R}$ *and* $P := \{x \,|\, Ax \leq b\}$ *is a polytope. Then* $\text{conv}(S)$ *is SOCr.*

Notice that we make no assumption regarding the structure or coefficients of the quadratic equation defining S. We require P to be a bounded polyhedron, which is not very restrictive given that in global optimization the variables are often assumed to be bounded to use branch-and-bound algorithms.

The result presented in Theorem 4.1 is somewhat unexpected since the sum-of-squares approach would build a sequence of SDP relaxations for (4.1) in order to optimize (exactly) a linear function over S, while even the SDP cone of three-by-three dimensional matrices is not SOCr [59]. Note that optimizing a linear function over S is NP-hard, therefore, while the convex hull is SOCr, the construction involves the introduction of an exponential number of variables.

Surprisingly, the proof of Theorem 4.1 is fairly straightforward and it introduces a technique (new, to the best of our knowledge) to compute convex hull of certain surfaces over a compact set. In the case of Theorem 4.1, the key observation is that the surface defined by the quadratic equation either:

1. is defined as the union of two convex surfaces (see Figure 4.1); or

2. it has the property that, through every point of the surface, there exists a *straight* line that is *entirely* contained in the surface (see Figure 4.2).

In Case 1, we can easily obtain that the convex hull of S is SOCr as we show in Section 4.3.3. In Case 2, no point in the interior of the polytope can be an extreme point of S. Observing that the convex hull of a compact set is also the convex hull of its extreme

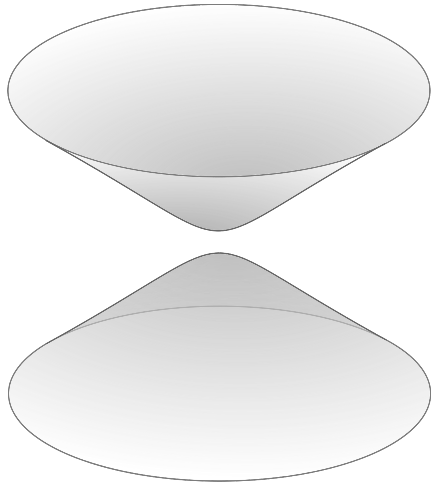

Figure 4.1: Two-sheets hyperboloid. The surface is the union of two convex peices.

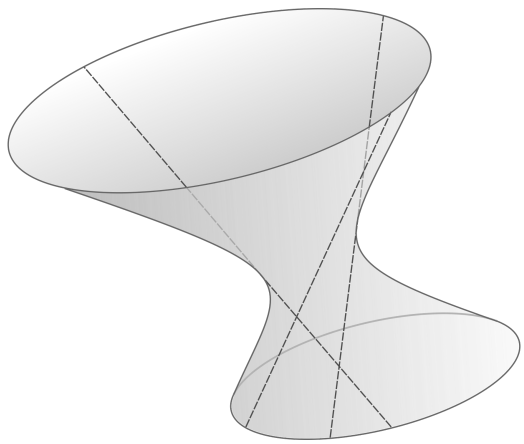

Figure 4.2: One-sheet hyperboloid. Through every point of the surface, there exists a straight line that is entirely contained in the surface.

points, we intersect the surface with each facet of the polytope which will contain all the extreme points of S. Now, each such intersection leads to new sets with the same form as S but in one dimension lower. The argument then goes by recursion. The details of the proof are presented in Section 4.3.

After we had proved Theorem 4.1, we learned that the property described in Case 2 is known as "ruled surfaces" and it has been extensively studied from both algebraic and geometric perspectives [57]. To the best of our knowledge, however, no one from the global optimization community has ever exploited such results for convexification.

4.3 Proof of Theorem 4.1

4.3.1 Convex hulls via disjunctions

In this section, we describe a simple procedure to obtain the convex hull of a compact set S using a disjunctive argument. We use this procedure to prove Theorem 4.1 in Section 4.3.3. Let S be a compact set and let extr(S) be the set of extreme points of S. First, we partition the extreme points of S. Specifically, suppose there exist $B^1, \ldots, B^k \subseteq S$ such that:

$$S \supseteq \bigcup_{i=1}^{k} B^i \supseteq \text{extr}(S). \tag{4.2}$$

We observe that (4.2) implies that

$$\text{conv}\,(S) \supseteq \text{conv}\left(\bigcup_{i=1}^{k} B^i\right) \supseteq \text{conv}\,(\text{extr}(S)) = \text{conv}\,(S)\,, \tag{4.3}$$

where the last equality holds due to S being compact. Finally, we obtain that

$$\text{conv}\,(S) = \text{conv}\left(\bigcup_{i=1}^{k} B^i\right) = \text{conv}\left(\bigcup_{i=1}^{k} \text{conv}\left(B^i\right)\right). \tag{4.4}$$

Observation 1. *If* conv(B^i) *is SOCr for all* $i \in [k]$*, then the set*

$$\text{conv}\left(\bigcup_{i=1}^{k} \text{conv}\left(B^i\right)\right),$$

is SOCr [19]. Thus, we obtain from (4.4) that conv(S) *is SOCr. In addition, we obtain a constructive procedure to compute* conv(S).

4.3.2 Reduction

In this section, we discuss how we can apply some transformations to the set S defined in (4.1) so as to re-write it in a "canonical" form where all the quadratic terms are squared terms. This will allows us to easily classify S into Case 1 and 2 as discussed in Section 4.2. We start with the following observation.

Observation 2. *Let $S \subseteq \mathbb{R}^n$ and let $F : \mathbb{R}^n \to \mathbb{R}^n$ be an affine map. Then*

$$\text{conv}(F(S)) = F(\text{conv}(S)),$$

where $F(S) := \{Fx \,|\, x \in S\}$. Furthermore if conv$(S)$ *is SOCr, then* conv$(F(S))$ *is also SOCr.*

Let S be the set defined in (4.1). Suppose, without loss of generality, that Q is a symmetric matrix. By the spectral theorem $Q = V^\top \Sigma V$, where Σ is a diagonal matrix and the columns of V are a set of orthogonal vectors. Letting $w = Vx$, we have that

$$S := V^{-1}\left(\{w \,|\, w^\top \Sigma w + \alpha^\top V^{-1} w = d,\ w \in \tilde{P}\}\right),$$

where $\tilde{P} := \{w \,|\, AV^{-1}w \leq b\}$.

Therefore, by Observation 2, it is sufficient to study the convex hull of a set of the form:

$$S := \left\{(x, y, z) \in \mathbb{R}^n \,|\, \sum_{i=1}^{n_q} a_i x_i^2 + \sum_{i=1}^{n_q} \alpha_i x_i + \sum_{j=1}^{n_l} \beta_j y_j = g,\ (x, y, z) \in P\right\},$$

where $z \in \mathbb{R}^{n_o}$ does not appear in the quadratic constraints, $n_q + n_l + n_o = n$, $a_i \neq 0$ for $i \in [n_q]$ (i.e., the rank of Q is n_q) and $\beta_j \neq 0$ for $j \in [n_l]$. By completing squares, we may further write S as:

$$\begin{aligned} S := \{(x, y, z) \in \mathbb{R}^n \,|\, &\sum_{i=1}^{n_q} \sigma(a_i)\left(\sqrt{|a_i|}x_i + \sigma(a_i)\frac{\alpha_i}{2\sqrt{|a_i|}}\right)^2 + \\ &\sum_{i=1}^{n_l} \beta_i y_i = g + \sum_{i=1}^{n_q} \frac{\alpha_i^2}{4a_i},\ (x, y, z) \in P\}, \end{aligned}$$

where $\sigma(a)$ denotes the sign of a. Now, since $u_i = \left(\sqrt{|a_i|}x_i + \sigma(a_i)\frac{\alpha_i}{2\sqrt{|a_i|}}\right)$ for $i \in [n_q]$

and $v_i = \beta_i y_i$ for $i \in [n_l]$ define linear bijections, it follows from Observation 2 that it is sufficient to study the convex hull of the following set:

$$S := \{(w, x, y, z) \in \mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_{q-}} \times \mathbb{R}^{n_l} \times \mathbb{R}^{n_o} \mid \sum_{i=1}^{n_{q+}} w_i^2 - \sum_{j=1}^{n_{q-}} x_j^2 + \sum_{k=1}^{n_l} y_k = g,$$
$$(w, x, y, z) \in P\}, \tag{4.5}$$

where we may further assume that $g \geq 0$, since otherwise we may multiply the equation by -1 and apply suitable affine transformations to bring it back to the form of (4.5).

4.3.3 Recursive argument to prove Theorem 4.1

We begin by stating a variant of Observation 2 that we will use twice along the proof.

Lemma 14. *Let $G = \{(x, w) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid x \in G_0,\ w = C^\top x + h\}$, where $G_0 \subseteq \mathbb{R}^{n_1}$ is bounded, and $C^\top x + h$ is an affine function of x. Then,*

$$\text{conv}(G) = \{(x, w) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid x \in \text{conv}(G_0),\ w = C^\top x + h\}.$$

Proof. See Lemma 4 in [55]. □

Dealing with low dimensional polytope

Let S and P be defined as in (4.1). Next, we show that we may assume without loss of generality that P is full dimension. In fact, if P is not full dimensional, then P is contained in a non-trivial affine subspace defined by a system of linear equations $Mx = f$. Without loss of generality, we may assume that M has full row-rank k, $1 \leq k < n$. Let $M = \begin{bmatrix} M_B & M_N \end{bmatrix}$ where M_B is invertible. Then, we may write this system as $x_B = -M_B^{-1} M_N x_N + M_B^{-1} f$, where $x_B \in \mathbb{R}^k,\ x_N \in \mathbb{R}^{n-k}$ and, for simplicity, we assume that x_B (resp. x_N) correspond to the first k (resp. last $n - k$) components of x. By defining

$C = -M_B^{-1} M_N$ and $h = M_B^{-1} f$ to simplify notation, we obtain

$$x_B = C x_N + h. \tag{4.6}$$

By partitioning Q in sub-matrices of appropriate sizes, we may explicitly write the quadratic equation defining S in terms of x_B and x_N as follows:

$$\begin{bmatrix} x_B^\top & x_N^\top \end{bmatrix} \begin{bmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{bmatrix} \begin{bmatrix} x_B \\ x_N \end{bmatrix} + \alpha^\top \begin{bmatrix} x_B \\ x_N \end{bmatrix} = g. \tag{4.7}$$

Using (4.6), we replace x_B in (4.7) to obtain

$$x_N^\top \tilde{Q} x_N + \tilde{\alpha}^\top x_N = \tilde{g},$$

where

$$\begin{aligned}
\tilde{Q} &= C^\top Q_{BB} C + C^\top Q_{BN} + Q_{NB} C + Q_{NN}, \\
\tilde{\alpha} &= 2C^\top Q_{BB} h + Q_{BN}^\top h + Q_{NB} h + C^\top \alpha_B + \alpha_N, \\
\tilde{g} &= g - h^\top Q_{BB} h - \alpha_B^\top h.
\end{aligned}$$

Therefore, we may write S as

$$S := \{(x_B, x_N) \in \mathbb{R}^n \mid \; x_N^\top \tilde{Q} x_N + \tilde{\alpha}^\top x_N = \tilde{g}, \; x_N \in \tilde{P}, \; x_B = C x_N + h\}, \tag{4.8}$$

where $\tilde{P}$ is now a full dimensional polytope. Therefore, by Lemma 14, we may assume from now on that P is full dimensional.

Case 2: Sufficient conditions for points to not be extreme

Consider the set S as defined in (4.5).

Lemma 15. *Suppose $n_o \geq 1$. If $(a, b, c, d) \in S \cap (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_{q-}} \times \mathbb{R}^{n_l} \times \mathbb{R}^{n_o})$ where $(a, b, c, d) \in \text{int}(P)$, then (a, b, c, d) is not an extreme point of S.*

Proof. Since $(a, b, c, d) \in \text{int}(P)$, there exists a vector $\delta \in \mathbb{R}^{n_o} \setminus \{0\}$ such that $(a, b, c, d + \delta), (a, b, c, d - \delta) \in P$. Clearly these points are in S as well and, therefore, (a, b, c, d) is not an extreme point of S □

Lemma 16. *Suppose $n_o = 0$ and $n_l \geq 2$. If $(a, b, c) \in S \cap (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_{q-}} \times \mathbb{R}^{n_l})$ where $(a, b, c) \in \text{int}(P)$, then (a, b, c) is not an extreme point of S.*

Proof. Since $n_l \geq 2$, $(a, b, c_1 \pm \lambda, c_2 \mp \lambda, \ldots, c_{n_3})$ are feasible for sufficiently small positive values of λ. Therefore, (a, b, c) is not an extreme point. □

Lemma 17. *Suppose $n_o = 0$, $n_{q+}, n_{q-} \geq 1$ and $n_l = 1$. If $(a, b, c) \in S \cap (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_{q-}} \times \mathbb{R}^{n_l})$ where $(a, b, c) \in \text{int}(P)$, then (a, b, c) is not an extreme point of S.*

Proof. Since $n_{q+}, n_{q-} \geq 1$, and $n_l = 1$, $(a_1 + \lambda, a_2, \ldots, a_{n_{q+}}, b_1 + \lambda, b_2, \ldots, b_{n_{q-}}, c + 2\lambda(-a_1 + b_1)$ are feasible for sufficiently small positive and negative values of λ. Therefore, (a, b, c) is not an extreme point. □

Lemma 18. *Suppose $n_o = 0$, $n_{q+} \geq 2$, $n_{q-} \geq 1$ and $n_l = 0$. If $(a, b) \in S \cap (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_{q-}})$ where $(a, b) \in \text{int}(P)$, then (a, b) is not an extreme point of S.*

Proof. We show that there exists a straight line through (a, b) that is entirely contained in the surface defined by the quadratic equation. More specifically, we prove that there exists a vector $(u, v) \in (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_{q-}}) \setminus \{0\}$ such that the line $\{(a, b) + \lambda(u, v) \mid \lambda \in \mathbb{R}\}$ satisfies the quadratic equation and therefore, (a, b) being in the interior of P cannot be an extreme point of S. We consider two cases:

1. $(a, b) \neq \mathbf{0}$: Then observe that $a \neq \mathbf{0}$, since otherwise we would have $a = \mathbf{0}$ and $b = \mathbf{0}$, because $g \geq 0$. Observe that

$$\sum_{i=1}^{n_{q+}} a_i^2 = g + \sum_{j=1}^{n_{q-}} b_j^2 \geq b_1^2 \Leftrightarrow \frac{|b_1|}{\|a\|_2} \leq 1. \tag{4.9}$$

Next, observe that:

$$\begin{aligned}
g &= \sum_{i=1}^{n_{q+}}(a_i+\lambda u_i)^2 - \sum_{i=1}^{n_{q-}}(b_i+\lambda v_i)^2 \;\forall\lambda\in\mathbb{R} \\
\Leftrightarrow g &= \left(\sum_{i=1}^{n_{q+}} a_i^2 - \sum_{i=1}^{n_{q-}} b_i^2\right) + \lambda^2\left(\sum_{i=1}^{n_{q+}} u_i^2 - \sum_{i=1}^{n_{q-}} v_i^2\right) \\
&\quad +2\lambda\left(\sum_{i=1}^{n_{q+}} a_i u_i - \sum_{i=1}^{n_{q-}} b_i v_i\right)\;\forall\lambda\in\mathbb{R} \\
\Leftrightarrow 0 &= \lambda\left(\sum_{i=1}^{n_{q+}} u_i^2 - \sum_{i=1}^{n_{q-}} v_i^2\right) + 2\left(\sum_{i=1}^{n_{q+}} a_i u_i - \sum_{i=1}^{n_{q-}} b_i v_i\right)\;\forall\lambda\in\mathbb{R} \\
\Leftrightarrow &\quad \sum_{i=1}^{n_{q+}} u_i^2 - \sum_{i=1}^{n_{q-}} v_i^2 = 0,\; \sum_{i=1}^{n_{q+}} a_i u_i - \sum_{i=1}^{n_{q-}} b_i v_i = 0. \qquad (4.10)
\end{aligned}$$

Suppose we set $v_1 = 1$ and $v_j = 0$ for all $j \in \{2, \ldots, n_{q-}\}$. Then satisfying (4.10) is equivalent to finding real values of u satisfying:

$$\sum_{i=1}^{n_{q+}} u_i^2 = 1,\;\; \sum_{i=1}^{n_{q+}} a_i u_i = b_1.$$

This is the intersection of a circle of radius 1 in dimension two or higher (since $n_{q+} \geq 2$ in this case) and a hyperplane whose distance from the orgin is $\frac{|b_1|}{\|a\|_2}$. Since, by (4.9), we have that this distance is at most 1, the hyperplane intersects the circle and therefore we know that a real solution exists.

2. $(a, b) = \mathbf{0}$: In this case, observe that $g = 0$ and then $\mathbf{0}$ is a convex combination of

$$(\;\underbrace{\pm\lambda, 0, \ldots, 0}_{\text{first } n_{q+} \text{ components}}\;,\;\underbrace{\pm\lambda, 0, \ldots, 0}_{\text{second } n_{q-} \text{ components}}\;)$$

for sufficiently small $\lambda > 0$.

□

Case 1: Sufficient conditions for convex hull to be SOCr

In this section, we repeatedly use the following result from [133].

Theorem 4.2. *Let $G \subseteq \mathbb{R}^n$ be a convex set and let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous function. Then*

$$\text{conv}\left(\{G \cap \{x \mid f(x) = 0\}\}\right) = \text{conv}\left(\{G \cap \{x \mid f(x) \leq 0\}\}\right) \cap \text{conv}\left(\{G \cap \{x \mid f(x) \geq 0\}\}\right).$$

For the two lemmas that follows, consider the notation of S defined in (4.5).

Lemma 19. *Suppose $n_o = 0$, $n_l \leq 1$. If $n_{q+} = 0$ or $n_{q-} = 0$, then* $\text{conv}(S)$ *is SOCr.*

Proof. We consider two cases.

1. $n_{q-} = 0$: Let $(w, y) \in S \cap (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_l})$. Let $y = y_1$ if $n_l = 1$ and $y = 0$ if $n_l = 0$. In this case, $g - y$ is non-negative for all feasible values of y and we can use the identity $t = \frac{(t+1)^2 - (t-1)^2}{4}$ to write $S = S' \cap S''$, where:

$$S' := \{(w, y) \in P \mid \|2w_1, \dots, 2w_{n_{q+}}, (g - y - 1)\| \leq (g - y + 1)\},$$
$$S'' := \{(w, y) \in P \mid \|2w_1, \dots, 2w_{n_{q+}}, (g - y - 1)\| \geq (g - y + 1)\}.$$

   Notice that S' is a SOCr convex set. Also notice that S'' is a reverse convex set intersected with a polytope and hence $\text{conv}(S'' \cap P)$ is polyhedral and contained in P (see [72],Theorem 1). Therefore, by Theorem 4.2, we have that $\text{conv}(S) = \text{conv}(S') \cap \text{conv}(S'')$ is SOCr.

2. $n_{q+} = 0$: Let $(x, y) \in S \cap (\mathbb{R}^{n_{q+}} \times \mathbb{R}^{n_l})$. Let $y = y_1$ if $n_l = 1$ and $y = 0$ if $n_l = 0$. In this case, $g - y$ is non-positive for all feasible values of y and may write $S = S' \cap S''$,

where:

$$S' := \{(x,y) \in P \,|\, \|2x_1, \ldots, 2x_{n_{q-}}, (y-g-1)\| \leq (y-g+1)\},$$
$$S'' := \{(x,y) \in P \,|\, \|2x_1, \ldots, 2x_{n_{q-}}, (y-g-1)\| \geq (y-g+1).\}$$

Therefore, as in the previous case, $\operatorname{conv}(S)$ is SOCr.

$\square$

Lemma 20. *Suppose* $n_{q+} \leq 1$ *and* $n_l = n_o = 0$. *Then* conv(S) *is SOCr.*

Proof. If $n_{q+} = 0$, then S is empty set or contains a single point, the origin.

Therefore, consider the case where $n_{q+} = 1$, thus $w = w_1$. Notice that $S = S' \cap S''$, where

$$S' := \{(w,x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid \; w^2 \geq g + \sum_{j=1}^{n_{q-}} x_j^2, \; (w,x) \in P\},$$
$$S'' := \{(w,x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid \; w^2 \leq g + \sum_{j=1}^{n_{q-}} x_j^2, \; (w,x) \in P\}.$$

By Theorem 4.2, conv$(S) =$ conv$(S') \cap$ conv(S''). Next, we show that both conv(S') and conv(S'') are SOCr. Notice that S' is the union of the following two SOCr sets:

$$S'_+ := \left\{(w,x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid \; w \geq \left(g + \sum_{j=1}^{n_{q-}} x_j^2\right)^{\frac{1}{2}}, \; w \geq 0, \; (w,x) \in P\right\},$$
$$= \text{Proj}_{w,x}\left(\left\{(w,x,t) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \times \mathbb{R} \,|\, \begin{array}{c} w \geq \left((\sqrt{g}t)^2 + \sum_{j=1}^{n_{q-}} x_j^2\right)^{\frac{1}{2}}, \\ x \geq 0, \; t = 1, \; (w,x) \in P, \end{array}\right\}\right)$$
$$S'_- := \left\{(w,x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid \; -w \geq \left(g + \sum_{j=1}^{n_{q-}} x_j^2\right)^{\frac{1}{2}}, \; w \leq 0, \; (w,x) \in P\right\}$$
$$= \text{Proj}_{w,x}\left(\left\{(w,x,t) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \times \mathbb{R} \,|\, \begin{array}{c} -w \geq \left((\sqrt{g}t)^2 + \sum_{j=1}^{n_{q-}} x_j^2\right)^{\frac{1}{2}}, \\ w \leq 0, \; t = 1, \; (w,x) \in P \end{array}\right\}\right).$$

Thus, $\operatorname{conv}(S') = \operatorname{conv}(S'_+ \cup S'_-)$ is SOCr.

Notice that $S'' = \{(w, x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid |w| \leq (g + \sum_{j=1}^{n_{q-}} x_j^2)^{\frac{1}{2}},\ (w, x) \in P\}$ and is therefore the union of two sets:

$$S''_+ := \left\{ (w,x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid w \leq \left(g + \sum_{j=1}^{n_{q-}} x_j^2 \right)^{\frac{1}{2}},\ w \geq 0,\ (w,x) \in P \right\},$$
$$S''_- := \left\{ (w,x) \in \mathbb{R}^1 \times \mathbb{R}^{n_{q-}} \mid -w \leq \left(g + \sum_{j=1}^{n_{q-}} x_j^2 \right)^{\frac{1}{2}},\ w \leq 0,\ (w,x) \in P \right\},$$

each of them being a reverse convex set intersected with a polyhedron. Therefore, $\operatorname{conv}(S''_+)$ and $\operatorname{conv}(S''_-)$ are polyhedral and therefore $\operatorname{conv}(S'') = \operatorname{conv}(\operatorname{conv}(S''_+) \cup \operatorname{conv}(S''_-))$ is a polyhedral set. $\square$

Proof of Theorem 4.1

Finally, we bring the pieces together to prove Theorem 4.1.

Proof. (of Theorem 4.1) Let $S(n)$ be defined as in (4.5), where $n = n_{q+} + n_{q-} + n_l + n_o$ is the dimension of the space in which S is defined and without loss of generality P is full-dimensional (Section 4.3.3). The proof goes by induction on n. Notice that $S(1)$ is a polytope and hence $\operatorname{conv}(S(1))$ is SOCr. Suppose $S(n)$ is SOCr. We show that $S(n+1)$ is SOCr as well. If $n_o = 0$, $n_l \leq 1$, and $n_{q+} = 0$ or $n_{q-} = 0$, then the result follows from Lemma 19. Similarly, if $n_o = 0$, $n_{q+} \leq 1$ and $n_l = 0$, then the result follows from Lemma 20. Otherwise, it follows from Lemma 15, 16, 17 and 18 that no point in the interior of P can be an extreme point of $S(n+1)$. Let N be the number of facets of P, each of which given by one equation of the linear system $Fx = f$. Let $B^i = S(n+1) \cap \{x \in \mathbb{R}^{n+1} \,|\, F_{i.}x = f_i\}$ be the intersection of $S(n+1)$ with the ith facet of P. By the discussion in Section 4.3.1, it is enough to show that the convex hull of each B^i is SOCr. Let $i \in \{1, \ldots, N\}$. Choose j_0 such that $F_{ij_0} \neq 0$. For simplicity, suppose $j_0 = 1$. Then, we may write $B^i = \{x \in \mathbb{R}^{n+1} \,|\, (x_2, \ldots, n_{n+1}) \in B_0^i,\ x_1 = b_i - \sum_{j=2}^{n+1} F_{ij} x_j\}$, where B^i_0 is obtained from B^i by replacing $x_1 = f_i - \sum_{j=2}^{n+1} F_{ij}x_j$ in all the constraints defining $S(n+1)$. Now $\text{conv}(B^i_0) \subseteq \mathbb{R}^n$ is SOCr by induction hyptothesis. Therefore, $\text{conv}(B^i)$ is SOCr by Lemma 14. $\square$

CHAPTER 5

A STUDY OF RANK-ONE SETS WITH LINEAR SIDE CONSTRAINTS AND APPLICATION TO THE POOLING PROBLEM

The work presented in this chapter has already been submitted. See first version of the paper at: `http://www.optimization-online.org/DB_HTML/2019/02/7056.html`.

5.1 Introduction

5.1.1 Motivation

A general quadratically constrained quadratic program (QCQP) is an optimization problem of the following form:

$$\begin{aligned} \min \quad & x^\top Q^0 x + (a^0)^\top x \\ \text{s.t.} \quad & x^\top Q^k x + (a^k)^\top x \le b_k \qquad \forall k \in \{1, \dots, m\} \\ & x \in [0,\ 1]^n, \end{aligned} \tag{5.1}$$

where the matrices Q^i for $i \in \{0, \dots, m\}$ are not assumed to be positive semidefinite.

Building convex relaxations of the feasible region of a QCQP is a key direction of research. Many general-purpose convexification schemes have been proposed for QCQPs. It includes, for example, the Reformulation-Linearization Technique (RLT) [127], the Lasserre hierarchy [86], and linear programming (LP) and second-order cone programming (SOCP) based alternatives to sum of squares optimization [3]. An important area of study regarding convexification schemes for QCQPs is to convexify commonly occurring substructures, like in the case of integer programming. However, most of the work in this direction in the global optimization area has been focused on convexification of functions (i.e., finding convex and concave envelopes), see for example [4, 120, 88, 20, 97, 18, 14, 100, 49, 126, 118, 99, 136, 137, 33, 44, 1, 68]. There are relatively lesser number of results on convexification of sets [135, 110, 111, 134, 66, 82, 115, 48, 87, 34, 104, 55]. It is well-known that it is possible to obtain tighter convex relaxations when convexifing a set directly rather than using convex envelopes of functions describing the set. In this chapter, we pursue the convexification of sets that appear as substructures of general QCQPs.

A common approach to obtain convex relaxations of QCQPs is that of using semidefinite programming (SDP) relaxations. The first step in this approach is to write an equivalent form of the QCQP (5.1) as follows:

$$\begin{aligned}
\min \quad & \langle Q^0, X\rangle + (a^0)^\top x && (5.2)\\
\text{s.t.} \quad & \langle Q^k, X\rangle + (a^k)^\top x \le b_k \qquad \forall k \in \{1,\ldots,m\} && (5.3)\\
& \operatorname{rank}\left(\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix}\right) = 1 && (5.4)\\
& x \in [0,\ 1]^n, && (5.5)
\end{aligned}$$

where $\langle U, V\rangle := \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} U_{ij}V_{ij}$. Observe that all the non-convexity of the problem is now captured by the rank-1 condition. This motivates our study of sets defined by a rank-1 [1] constraint together with some linear side constraints:

$$\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]_{k=1}^m) := \left\{W \in \mathbb{R}_+^{n_1\times n_2} \mid \langle A^k, W\rangle \le b_k, \forall k \in \{1,\ldots,m\},\ \operatorname{rank}(W) \le 1\right\} \quad (5.6)$$

where we will recover (5.3)–(5.5) if we replace W with $\begin{bmatrix} 1 & x^\top \\ x & X \end{bmatrix}$ and with appropriate choice of A^ks.

The starting point of our investigations is the classical result of [36] that says: if the linear inequalities in (5.6) are $W_{ij} \le 1$ for all $i \in \{1,\ldots,n_1\}$ and $j \in \{1,\ldots,n_2\}$, then $\operatorname{conv}(\mathcal{U})$ is exactly the boolean quadric polytope [114]. This is a well-studied polytope

[1] With some abuse of terminology, we will be referring to matrices whose rank is at most one by simply rank-1 matrices.

and inequalities describing this set, such as the McCormick inequalities [95] and triangle inequalities [114], are already used in practice. The paper [30] shows the use of more complicated inequalities valid for the boolean quadric polytope for solving box-constrained quadratic programs.

However, to the best our knowledge, no other particular choice of structured linear side constraint has ever been studied (see [23] for intersection cuts for rank-1 sets with general linear constraints). Let us give a simple choice of linear side constraints as a motivating example (where we do not assume the matrix variable is a square matrix): bounds on rows or columns of the W variable, i.e., the set

$$\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u) := \left\{ W \in \mathbb{R}^{n_1 \times n_2}_{+} \mid l_i \leq \sum_{j=1}^{n_2} W_{ij} \leq u_i, \forall i \in \{1,\ldots,n_1\},\ \operatorname{rank}(W) \leq 1 \right\} \tag{5.7}$$

where we assume $0 \leq l \leq u$. This choice of side constraints is not arbitrary, but comes up naturally for the pooling problem [71, 67]. Also note that such a relaxation could always be constructed for any bounded QCQP.

5.1.2 Contributions

In this chapter, we explore general conditions under which the convex hull of the set $\mathcal{U}^{m}_{(n_1,n_2)}([A^k, b_k]_{k=1}^{m})$ is polyhedral or second-order cone representable (SOCr), and also show that in each of these cases a linear objective function can be optimized over these sets in polynomial time. These results are presented in Section 5.2. It turns out that the set $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$ introduced in (5.7) is a special case of the sets studied in Section 5.2 and its convex hull is polyhedral.

In Section 5.3, we specialize the general results of Section 5.2 for sets like $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$ that are applicable for the pooling problem. We present results on the polyhedrality of convex hull, valid inequalities and extended formulations, and complexity of separating inequalities in the original space. Specifically in Section 5.3.3, we present several formulations (and related discretizations) of the generalized pooling problem (i.e., pooling problems that have pool-to-pool arcs). Then, we illustrate how the sets studied here appear as substructures in different ways in the different pooling formulations.

Finally, in Section 5.4, we present results from computational experiments, which show that the new inequalities generated from substructures similar to $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$, help in improving dual bounds significantly.

5.2 General results

Notation: We use $\text{conv}(S)$ to denote convex hull of a set S, $\text{extr}(S)$ to denote the set of extreme points of a set S, $\text{proj}_x(S)$ to denote the projection of a set S onto the x variables, $[n]$ to denote the set $\{1,\ldots,n\}$, and $\mathbb{R}^n_{++}$ to be the set of n dimensional positive vectors. Let us also define the set of $m-$partitions of a set S as

$$\mathcal{P}_m(S) := \{(S_1,\ldots,S_m) \mid \cup_{k=1}^m S_k = S,\ S_j \cap S_k = \emptyset \text{ for } j \neq k\}.$$

In this section, we present cases in which $\text{conv}\left(\mathcal{U}^m_{(n_1,n_2)}([A^k,b_k]_{k=1}^m)\right)$ is either polyhedral or second-order cone representable.

5.2.1 Some cases with polyhedral representable convex hulls

We start with the trivial case of a single linear side constraint.

Proposition 13. *Suppose the set* $\mathcal{U}^1_{(n_1,n_2)}([A^1,b_1])$ *defined as in* (5.6) *is bounded*[2]*. Then, we have*

$$\text{conv}\left(\mathcal{U}^1_{(n_1,n_2)}([A^1,b_1])\right) := \left\{W \in \mathbb{R}^{n_1\times n_2}_+ \mid \langle A^1, W\rangle \le b_1\right\}.$$

The proof of Proposition 13 follows from the fact that the extreme points of the set

$$\left\{W \in \mathbb{R}^{n_1\times n_2}_+ \mid \langle A^1, W\rangle \le b_1\right\},$$

[2]Boundedness is equivalent to $A^1_{ij} > 0$ for all $i \in [n_1],\ j \in [n_2]$.

are rank-1 matrices. Therefore, convexifying with just one constraint is not very interesting.

Next, we allow multiple linear side constraints with certain rank-1 constraint matrices.

Theorem 5.1. *Consider the set* $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})$ *defined in* (5.6) *where the constraint matrices are of the form*

$$A^k := \alpha^k \beta^\top \qquad k \in [m],$$

where $\alpha^k \in \mathbb{R}^{n_1}$ *for* $k \in [m]$ *and* $\beta \in \mathbb{R}^{n_2}_{++}$*. Moreover, let the* α^k*'s be such that*

$$\{u \in \mathbb{R}^{n_1}_+ \,|\, (\alpha^k)^\top u \leq 0, \forall\, k \in [m]\} = \{0\}. \tag{5.8}$$

Then, the following hold:

1. $\operatorname{conv}\left(\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})\right)$ *is a polyhedral set.*

2. *The set of extreme points of* $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})$ *are of the form:*

$$\operatorname{extr}\left(\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})\right) = \left\{\gamma e_h^\top \,|\, h \in [n_2]\right\},$$

   where e_h *is the* n_2*-dimensional vector with all components zero except the* h^{th} *component which is* 1 *and the* γ*'s are extreme points of the set:*

$$\left\{\gamma \in \mathbb{R}^{n_1}_+ \,|\, \sum_{i=1}^{n_1} \alpha_i^k \beta_h \gamma_i \leq b_k,\ k \in [m]\right\}.$$

3. *A compact extended formulation of* $\operatorname{conv}\left(\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})\right)$ *is given by:*

$$\sum_{j=1}^{n_2} t_j = 1 \tag{5.9}$$

$$\sum_{i=1}^{n_1} \alpha_i^k \beta_j W_{ij} \leq b_k t_j \qquad \forall k \in [m], j \in [n_2] \tag{5.10}$$

$$t_j \geq 0 \qquad \forall j \in [n_2]. \tag{5.11}$$

Therefore, a linear function can be optimized in polynomial time on $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]_{k=1}^m)$.

Proof. To simplify notation, we will just write $\mathcal{U}^m_{(n_1,n_2)}$ instead of $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]_{k=1}^m)$ in this proof.

Observe first that condition (5.8) together with the fact that $\beta_j > 0$ for $j \in [n_2]$ imply that $\{W \in \mathbb{R}^{n_1 \times n_2}_+ \,|\, \langle \alpha^k \beta^\top, W\rangle \leq 0, \forall k \in [m]\}$ is bounded, and therefore $\mathcal{U}^m_{(n_1,n_2)}$ is bounded. Therefore, to prove (i), it is sufficient to show that the set of extreme points is finite. We will begin by showing that in any extreme point of the set $\mathcal{U}^m_{(n_1,n_2)}$, there is at most one non-zero column. By contradiction, let us assume that $\hat{W}$ is an extreme point with two non-zero columns. Since $\mathrm{rank}(\hat{W}) \leq 1$ and $\hat{W} \geq 0$, there exists two vectors $\hat{x} \in \mathbb{R}^{n_1}_+$ and $\hat{y} \in \mathbb{R}^{n_2}_+$ such that $\hat{W} = \hat{x}\hat{y}^\top$ (note we may also assume $\hat{x} \in \mathbb{R}^{n_1}_-$ and $\hat{y} \in \mathbb{R}^{n_2}_-$; however we cannot have $\hat{x}$ and $\hat{y}$ with different component with different signs, since then $\hat{W}$ is not non-negative). Without loss of generality, let us assume that the first two components of y are positive. Let us define the set $\mathcal{K}(W) := \{k : \langle A^k, W\rangle = b_k\}$. Consider two new points $W^{\pm} = \hat{x}(\hat{y} \pm \epsilon_1 e_1 \mp \epsilon_2 e_2)^\top$. Observe that we can select small but positive ϵ_1 and ϵ_2 such that both W^+ and W^- belong to $\mathcal{U}^m_{(n_1,n_2)}$ since for each $k \in \mathcal{K}(\hat{W})$ we have

$$b_k = \langle \alpha^k \beta^\top, \hat{x}(\hat{y} \pm \epsilon_1 e_1 \mp \epsilon_2 e_2)^\top\rangle \Leftrightarrow (\hat{x}^\top \alpha^k)(\beta_1 \epsilon_1 - \beta_2 \epsilon_2) = 0 \Leftrightarrow \beta_1 \epsilon_1 = \beta_2 \epsilon_2,$$

where note that $\beta_1 > 0$ and $\beta_2 > 0$. Hence, we reach a contradiction to $\hat{W}$ being an extreme point.

Finally, the fact that there is at most one non-zero column in an extreme point of $\mathcal{U}^m_{(n_1,n_2)}$, the extreme points with j^{th} column being non-zero is of the form:

$$W^u = 0, u \in [n_2] \setminus \{j\},\ W \geq 0,\ \beta_j (\alpha^k)^\top W^j \leq b_k\ \forall k \in [m], \tag{5.12}$$

where W^u is the u^{th} column of W. Thus, there are finitely many extreme points. Hence, the result follows. This also proves (ii).

To prove (iii), let $V^m_{(n_1,n_2)}$ be the compact extended formulation defined in (5.9)-(5.11).

$\text{conv}\left(\mathcal{U}^m_{(n_1,n_2)}\right) \subseteq \text{proj}_W\left(V^m_{(n_1,n_2)}\right)$: Part (ii) lists all the extreme points of $\mathcal{U}^m_{(n_1,n_2)}$. It is straightforward to check that these extreme points belong to $\text{proj}_W\left(V^m_{(n_1,n_2)}\right)$.

$\text{proj}_W\left(V^m_{(n_1,n_2)}\right) \subseteq \text{conv}\left(\mathcal{U}^m_{(n_1,n_2)}\right)$: We first claim that in any extreme point of $V^m_{(n_1,n_2)}$ exactly one t variable is positive. In particular, let $(\hat{t}, \hat{W}) \in V^m_{(n_1,n_2)}$ and without loss of generality let $\hat{t}_1 \neq 0, \hat{t}_2 \neq 0$. Let ϵ such that $0 < \epsilon \leq \frac{1}{2} \cdot \min\{\hat{t}_1, \hat{t}_2\}$. Then we construct two solutions $(\tilde{t}, \tilde{W})$ and $(\bar{t}, \bar{W})$ as follows:

$\tilde{t}_1 = \hat{t}_1 - \epsilon, \bar{t}_1 = \hat{t}_1 + \epsilon; \tilde{t}_2 = \hat{t}_1 + \epsilon, \bar{t}_1 = \hat{t}_1 - \epsilon; \tilde{t}_j = \bar{t}_j = \hat{t}_j \forall j \in [n_2] \setminus \{1, 2\}$;

For $j \in \{1, 2\}$: $\tilde{W}^{ij} = \frac{\tilde{t}_j}{\hat{t}_j}\hat{W}_{ij}$ and $\bar{W}^{ij} = \frac{\bar{t}_j}{\hat{t}_j}\hat{W}_{ij}$; $\tilde{W}_{ij} = \bar{W}_{ij} = \hat{W}_{ij} \forall i \in [n_1], j \in [n_2] \setminus \{1, 2\}$.

It is straighforward to verify that $(\tilde{t}, \tilde{W})$ and $(\bar{t}, \bar{W})$ belong to $V^m_{(n_1,n_2)}$. Thus, $(\hat{t}, \hat{W})$ is not an extreme point of $V^m_{(n_1,n_2)}$.

Since an extreme point of $V^m_{(n_1,n_2)}$ has exactly one positive component in t, each extreme point of $V^m_{(n_1,n_2)}$ are of the form: $t_{j'} = 1$, and $W_{j'}$ is an extreme point of

$$\left\{ x \in \mathbb{R}^{n_1}_+ \mid \sum_i \alpha^k_i \beta_{j'} x_i \leq b^k \right\},$$

for some j' and $t_j = 0$, $W_{ij} = 0$ for all $j \in [n_2] \setminus j'$ ($t_j = 0$ implies $W_{ij} = 0$, due to condition (5.8)). Thus $\text{proj}_W\left(\text{extr}\left(V^m_{(n_1,n_2)}\right)\right) = \text{extr}\left(\mathcal{U}^m_{(n_1,n_2)}\right)$. Therefore, we obtain that $\text{proj}_W\left(V^m_{(n_1,n_2)}\right) \subseteq \text{conv}\left(\mathcal{U}^m_{(n_1,n_2)}\right)$. □

5.2.2 Some cases with second-order cone representable convex hulls

In this section, we study two cases in which the convex hull is SOCr but not necessary polyhedral. We first consider sets with two arbitrary linear side constraints.

Theorem 5.2. *Suppose that the set* $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]^2_{k=1})$ *as defined in* (5.6) *is bounded. Then, the convex hull of* $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]^2_{k=1})$ *is SOCr. Moreover, a linear function can be* *optimized in polynomial time over* $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]_{k=1}^2)$.

Proof. Note that the since $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]_{k=1}^2)$ is assumed to be bounded, we have that the convex hull of $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]_{k=1}^2)$ is the convex hull of its extreme points.

We first claim that in any extreme point, there are at most two non-zero columns. By contradiction, let us assume that $\hat{W}$ is an extreme point with three non-zero columns. Since $\mathrm{rank}(\hat{W}) = 1$ and $\hat{W} \geq 0$, there exists two vectors $\hat{x} \in \mathbb{R}^{n_1}_+$ and $\hat{y} \in \mathbb{R}^{n_2}_+$ such that $\hat{W} = \hat{x}\hat{y}^\top$. Without loss of generality, let us assume that the first three components of y are positive. Consider two new points $W^{\pm} = \hat{x}(\hat{y} \pm \epsilon_1 e_1 \pm \epsilon_2 e_2 \pm \epsilon_3 e_3)^\top$. Note that we can select a non-zero vector $(\epsilon_1, \epsilon_2, \epsilon_3)$ with small enough magnitude such that both W^+ and W^- belong to $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]_{k=1}^2)$ since we have

$$
\begin{aligned}
&\langle A^k, \hat{x}\hat{y}^\top \rangle = \langle A^k, \hat{x}(\hat{y} \pm \epsilon_1 e_1 \pm \epsilon_2 e_2 \pm \epsilon_3 e_3)^\top \rangle \\
&\qquad \Leftrightarrow 0 = (\hat{x}^\top A^k e_1)\epsilon_1 + (\hat{x}^\top A^k e_2)\epsilon_2 + (\hat{x}^\top A^k e_3)\epsilon_3 \;\; k = 1, 2.
\end{aligned}
$$

Note that this homogeneous system is guaranteed to have a non-trivial solution in $(\epsilon_1, \epsilon_2, \epsilon_3)$. Hence, we reach to a contradiction.

Following a similar procedure, one can also show that there are at most two non-zero rows in any extreme point. Therefore, we deduce that the largest non-zero submatrix of the extreme point of the set $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]_{k=1}^2)$ is 2×2.

Now, let us fix a particular two-by-two submatrix of W. We will show that the convex hull of all extreme points whose support is on this particular choice of two-by-two submatrix is SOCr. Since the convex hull of the union of SOCr compact sets is SOCr, we have that convex hull of the extreme points of $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]_{k=1}^2)$ is SOCr, which would complete the proof. Without loss of generality, consider the extreme points with support on the first two rows and first two columns. Then we are looking for the convex hull of the following set (which is satisfied by all the extreme points):

$$\begin{aligned}\langle A^k, W\rangle &\le b_k \ \forall k \in \{1,2\}\\ W_{ij} &\ge 0 \ \forall i \in [n_1], j \in [n_1]\\ W_{ij} &= 0 \ \text{ if } i \ge 3 \text{ or } j \ge 3\\ W_{11}W_{22} &= W_{21}W_{12}.\end{aligned} \tag{5.13}$$

It was recently shown in [55] that the convex hull of a set described by a quadratic constraint and bounded polyhedral constraints is SOCr (note that since (5.13) represents a subset of $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]^2_{k=1})$, it is bounded). This completes the proof.

Finally, note that optimizing a linear function on $\mathcal{U}^2_{(n_1,n_2)}([A^k, b_k]^2_{k=1})$ is equivalent to optimizing a linear function on the set of its extreme points. Since there are $\binom{n_1}{2} \cdot \binom{n_2}{2}$ possible choices of supports and for each choice, we can optimize a linear function over (5.13) in polynomial time (the size of the SOC-representation is linear in the total number of faces of the polytope, which is fixed in (5.13) since there are only six linear inequalities in four variables). $\square$

Note that the papers [34, 104] show that the convex hull of two quadratic constraints is SOCr. Although Theorem 5.2 is of similar flavor, it allows for the additional side constraints, namely, the non-negativities on the bilinear terms (i.e., the constraint $W \ge 0$). Moreover, the proof technique here is completely different from those used in [34, 104].

Next, we allow multiple linear side constraints with certain rank-2 constraint matrices and show that a result similar to Theorem 5.2 is possible.

Theorem 5.3. *Let* $n_1 = n_2 \ge 3$ *and suppose that the set* $\mathcal{U}^m_{(n_1,n_1)}([A^k, b_k]^m_{k=1})$ *defined in* (5.6) *is bounded. Assume that the constraint matrices are of the form*

$$A^k := \alpha^k \beta\beta^\top + \gamma^k \delta\delta^\top \qquad k \in [m],$$

where $\alpha^k, \gamma^k \in \mathbb{R}_+$ *for* $k \in [m]$ *and* $\beta, \delta \in \mathbb{R}^{n_1}_{++}$*. Then,* $\operatorname{conv}\left(\mathcal{U}^m_{(n_1,n_1)}([A^k, b_k]^m_{k=1})\right)$ *is*

SOCr. Moreover, for a fixed m, a linear function can be optimized in polynomial time over $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})$.

Proof. First, note that the non-negativity of α^k, γ^k and positivity of β, δ imply that the set is bounded.

Observe that the following system has a non-trivial solution in ϵ

$$\begin{bmatrix} \beta_{j_1} & \beta_{j_2} & \beta_{j_3} \\ \delta_{j_1} & \delta_{j_2} & \delta_{j_3} \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

for every distinct indices j_1, j_2, j_3. This guarantees that the following system has a non-trivial solution in ϵ:

$$\begin{aligned} \alpha^k(\beta^\top \hat{x})[\beta_{j_1}\epsilon_1 + \beta_{j_2}\epsilon_2 + \beta_{j_3}\epsilon_3] = 0 \quad k \in [m] \\ \gamma^k(\delta^\top \hat{x})[\delta_{j_1}\epsilon_1 + \delta_{j_2}\epsilon_2 + \delta_{j_3}\epsilon_3] = 0 \quad k \in [m]. \end{aligned} \tag{5.14}$$

After establishing the above relation, we proceed similar to the proof of Proposition 5.2. We again claim that in any extreme point, there are at most two non-zero columns. By contradiction, let us assume that $\hat{W} = \hat{x}\hat{y}^\top$ is an extreme point with three non-zero columns. Without loss of generality, let these be the first three columns. However, in this case, the following system has a non-trivial solution in ϵ,

$$\alpha^k(\beta^\top \hat{x})[\beta_1\epsilon_1 + \beta_2\epsilon_2 + \beta_3\epsilon_3] + \gamma^k(\delta^\top \hat{x})[\delta_1\epsilon_1 + \delta_2\epsilon_2 + \delta_3\epsilon_3] = 0 \quad k \in [m],$$

due to (5.14) (with the choice of $j_1 = 1, j_2 = 2, j_3 = 3$). Therefore, two points defined as $W^\pm = \hat{x}(\hat{y} \pm \epsilon_1 e_1 \pm \epsilon_2 e_2 \pm \epsilon_3 e_3)^\top$ belong to $\mathcal{U}^m_{(n_1,n_1)}([A^k, b_k]^m_{k=1})$ for some non-zero ϵ, contradicting to the assumption that $\hat{W}$ is an extreme point.

Using the same procedure, one can also prove that at an extreme point of $\operatorname{conv}\left(\mathcal{U}^m_{(n_1,n_1)}([A^k, b_k]^m_{k=1})\right)$, there can be at most two non-zero rows. Therefore, we deduce that the largest non-zero submatrix of an extreme point is 2×2. The rest of the proof is identical to the last part of the proof of Theorem 5.2. $\square$

5.3 Application of results to the pooling problem

In this section, we show how the results presented in Section 5.2 apply to the pooling problem.

5.3.1 Convex hull results and valid inequalities

Our first result of this section, Proposition 14 presented below, is regarding the convex hull of the set $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$ introduced in (5.7).

Proposition 14. *Consider the set $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$ described in (5.7) where we assume u_i is finite for every $i \in [n_1]$. Then, the following hold:*

1. $\text{conv}\left(\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)\right)$ *is a polyhedral set.*
2. *A compact extended formulation of* $\text{conv}\left(\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)\right)$ *is given by:*

$$\sum_{j=1}^{n_2} t_j = 1 \tag{5.15}$$

$$l_i t_j \leq W_{ij} \leq u_i t_j \qquad \forall i \in [n_1], j \in [n_2] \tag{5.16}$$

$$t_j \geq 0 \qquad \forall j \in [n_2]. \tag{5.17}$$

   Therefore, a linear function can be optimized in polynomial time on $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$.

3. *Let $\mathcal{I} := \{i \in [n_1] \,|\, l_i > 0\}$. Without loss of generality, let us assume that the first $|\mathcal{I}|$ components of l are positive. Assuming $u_i > 0$ for all $i \in [n_1]$ (otherwise, we may fix all the variable in the corresponding row to be zero),* $\text{conv}\left(\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)\right)$ *is given*

by:

$$\sum_{i=1}^{n_1} \sum_{j \in T_i} \frac{W_{ij}}{u_i} \leq 1 \qquad \forall (T_1, \ldots, T_{n_1}) \in \mathcal{P}_{n_1}([n_2]) \tag{5.18}$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in T_i} \frac{W_{ij}}{l_i} \geq 1 \qquad \forall (T_1, \ldots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}([n_2]) \tag{5.19}$$

$$l_{i_1} W_{i_2 j} \leq u_{i_2} W_{i_1 j} \qquad \forall j \in [n_2], i_1 \in \mathcal{I}, i_2 \in [n_1] \text{ and } i_1 \neq i_2 \tag{5.20}$$

$$W_{ij} \geq 0 \qquad \forall i \in [n_1], j \in [n_2], \tag{5.21}$$

where (5.19) are not required (in fact not well-defined) in the convex hull description if $\mathcal{I} = \emptyset$. The inequalities (5.18) and (5.19) can be separated in polynomial-time.

Note that parts (i) and (ii) of Proposition 14 are a corollary of Theorem 5.1, where we choose $\beta = e$ and, for $i \in [n_1],\ k \in [2n_1]$, we define: $\alpha^k = e_i,\ b_k = u_i$ if $k = i$; $\alpha^k = -e_i,\ b_k = -l_i$ if $k = n_1 + i$. For part (iii), we project the extended formulation presented in part (ii) to the space of the original variables W via Fourier-Motzkin procedure. The details of our proof of (iii) is presented in Appendix A.1.

We next highlight an interesting connection between the extended formulation presented in Proposition 14 and McCormick inequalities. In particular, note that since W is a rank-1 matrix, we have that $\frac{W_{ij}}{\sum_{j'} W_{ij'}}$ is a constant independent of the row index i (assuming $\sum_{j'} W_{ij'} \neq 0$). More formally, we may write:

$$W_{ij} = t_j \left(\sum_{j'=1}^{n_2} W_{ij'} \right) \qquad \forall i \in [n_1], j \in [n_2], \tag{5.22}$$

where the fact that the ratio variable t is independent of row index, is indicated with it being indexed only with column index j. Note that it is straightforward to see that the t variables satisfy:

$$\sum_{j=1}^{n_2} t_j = 1,\ t_j \geq 0 \qquad \forall j \in [n_2]. \tag{5.23}$$

Finally, we may apply McCormick inequalities for (5.22), by observing that we have the bounds $0 \leq t_j \leq 1$ and $l_i \leq \sum_{j'=1}^{n_2} W_{ij'} \leq u_i$, to obtain:

$$\begin{aligned} l_i t_j \leq \quad & W_{ij} & \leq u_i t_j \qquad (5.24)\\ u_i t_j + \sum_{j'=1}^{n_2} W_{ij'} - u_i \leq \quad & W_{ij} & \leq l_i t_j + \sum_{j'=1}^{n_2} W_{ij'} - l_i. \qquad (5.25)\end{aligned}$$

Now, observe that (5.23) together with the first two inequalities of the McCormick envelopes (5.24) above yields the extended formulation. Indeed, one can also check that (5.25) is implied by (5.24) and (5.23). This connection between McCormick inequalities and the extended formulation of the convex hull will be used later when we discretize various substructures of the pooling problem (see Section 5.3.2).

Next, we record another simple application of Theorem 5.1 that will be useful for the pooling problem. Consider the set

$$\begin{aligned} \mathcal{U}^{\text{row+}}_{(n_1,n_2)}(l,u,L,U) := \quad & \{W \in \mathbb{R}_+^{n_1 \times n_2} \mid \\ & l_i \leq \textstyle\sum_{j=1}^{n_2} W_{ij} \leq u_i, \qquad \forall i \in [n_1]\} \\ & L \leq \textstyle\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} W_{ij} \leq U \\ & \text{rank}(W) \leq 1\}, \end{aligned} \qquad (5.26)$$

where we assume $0 < u_i \leq U$ (otherwise we can replace u_i by U) and $U \leq \sum_{i=1}^{n_1} u_i$ (otherwise we replace U by $\sum_{i=1}^{n_1} u_i$).

Proposition 15. *Consider the set $\mathcal{U}^{\text{row+}}_{(n_1,n_2)}(l,u,L,U)$ described in (5.26) where we assume that U is finite. Then, the following hold:*

1. *An extended formulation of* $\operatorname{conv}\left(\mathcal{U}^{\text{row+}}_{(n_1,n_2)}(l,u,L,U)\right)$ *is given by:*

$$\sum_{j=1}^{n_2} t_j = 1 \tag{5.27}$$

$$Lt_j \leq \sum_{i=1}^{n_1} W_{ij} \leq Ut_j \qquad \forall j \in [n_2] \tag{5.28}$$

$$l_i t_j \leq W_{ij} \leq u_i t_j \qquad \forall i \in [n_1], \forall j \in [n_2] \tag{5.29}$$

$$t_j \geq 0 \qquad \forall j \in [n_2]. \tag{5.30}$$

2. *Let* $\mathcal{I} := \{i \in [n_1] \,|\, l_i > 0\}$*. Without loss of generality, let us assume that the first* $|\mathcal{I}|$ *components of* l *are positive. Assuming* $u_i > 0$ *for all* $i \in [n_1]$*,* $\operatorname{conv}\left(\mathcal{U}^{\text{row+}}_{(n_1,n_2)}(l,u,L,U)\right)$ *is given by:*

$$\sum_{i=1}^{n_1}\sum_{j\in T_i}\frac{W_{ij}}{u_i} + \frac{1}{U}\sum_{i=1}^{n_1}\sum_{j\in T_0} W_{ij} \leq 1 \qquad \forall (T_0, T_1, \ldots, T_{n_1}) \in \mathcal{P}_{n_1}([n_2]) \tag{5.31}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in T_i}\frac{W_{ij}}{l_i} + \frac{1}{L}\sum_{i\in\mathcal{I}}\sum_{j\in T_0} W_{ij} \geq 1 \qquad \forall (T_0, T_1, \ldots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}([n_2]) \tag{5.32}$$

(5.20) – (5.21).

The inequalities (5.31) and (5.32) can be separated in polynomial-time.

Part (i) of Proposition 15 is a corollary of Theorem 5.1. It can also be viewed as an extension of Proposition 14(ii), where we further define $\alpha^k = e,\ b^k = U$ if $k = 2n_1 + 1$; and $\alpha^k = -e,\ b^k = -L$ if $k = 2n_1 + 2$. The proof of part (ii) is presented in Appendix A.2.

Clearly, all the results above also apply to a set where instead of row bounds, we have column bounds, i.e., the set:

$$\mathcal{U}^{\text{col}}_{(n_1,n_2)}(l',u') := \left\{ W \in \mathbb{R}^{n_1\times n_2}_{+} \,|\, l'_j \leq \sum_{i=1}^{n_1} W_{ij} \leq u'_j, \forall j \in [n_2], \text{rank}(W) \leq 1 \right\}, \tag{5.33}$$

and the set $\mathcal{U}^{\text{col+}}_{(n_1,n_2)}(l',u',L,U)$ defined analogously to $\mathcal{U}^{\text{row+}}_{(n_1,n_2)}(l,u,L,U)$. A natural extension to these results is the intersection of the two sets. The first observation is that convex hull is not polyhedral.

Theorem 5.4. conv $\left(\mathcal{U}^{\text{row}}_{(2,2)}(l,u) \cap \mathcal{U}^{\text{col}}_{(2,2)}(l',u')\right)$ *is SOCr, but not polyhedral.*

A proof of Theorem 5.4 is presented in Appendix A.3. Unfortunately, we do not have a complete description of conv$(\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u) \cap \mathcal{U}^{\text{col}}_{(n_1,n_2)}(l',u'))$ for general $n_1, n_2 \in \mathbb{Z}_+$. We conjecture that optimizing a linear function on $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u) \cap \mathcal{U}^{\text{col}}_{(n_1,n_2)}(l',u')$ is NP-hard.

5.3.2 Relaxations and restrictions by discretization

We now provide inner and outer-approximations of the set $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$ via discretization. See [101, 67, 51] for more details and examples of application of this technique to the pooling problem. Let us start with the outer-approximation (or relaxation) and the discretization of the t_j variable as

$$t_j = \sum_{h=1}^{H} 2^{-h} z_{jh} + \gamma_j,$$

where $H \in \mathbb{Z}_{++}$ is the discretization level, z_{jh} are binary variables and γ_j is a continuous non-negative variable upper-bounded by 2^{-H}. As discussed in the previous section, applying McCormick inequalities to (5.22) yields the convex hull of $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$. Thus it makes sense to discretize t in this equation, i.e., we can now rewrite (5.22) as

$$W_{ij} = \left(\sum_{j'=1}^{n_2} W_{ij'}\right)\left(\sum_{h=1}^{H} 2^{-h} z_{jh} + \gamma_j\right) \qquad \forall i \in [n_1], \forall j \in [n_2].$$

Let us define $\alpha_{ijh} := \left(\sum_{j'=1}^{n_2} W_{ij'}\right) z_{jh}$ and $\beta_{ij} := \left(\sum_{j'=1}^{n_2} W_{ij'}\right)\gamma_j$, and write down the McCormick envelopes respectively as

$$l_i z_{jh} \leq \alpha_{ijh} \leq u_i z_{jh} \qquad \forall i \in [n_1], \forall j \in [n_2], \forall h \in [H] \quad (5.34)$$

$$u_i z_{jh} + \sum_{j'=1}^{n_2} W_{ij'} - u_i \leq \alpha_{ijh} \leq l_i z_{jh} + \sum_{j'=1}^{n_2} W_{ij'} - l_i \quad \forall i \in [n_1], \forall j \in [n_2], \forall h \in [H], \quad (5.35)$$

and

$$\begin{aligned} l_i\gamma_j \leq &\beta_{ij} \leq u_i\gamma_j && \forall i \in [n_1], \forall j \in [n_2] \quad (5.36) \\ u_i\gamma_j + \frac{1}{2^H}\sum_{j'=1}^{n_2} W_{ij'} - \frac{u_i}{2^H} \leq &\beta_{ij} \leq l_i\gamma_j + \frac{1}{2^H}\sum_{j'=1}^{n_2} W_{ij'} - \frac{l_i}{2^H} && \forall i \in [n_1], \forall j \in [n_2]. \quad (5.37) \end{aligned}$$

Then, we obtain the following outer-approximation of $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$:

$$\begin{aligned} \overline{\mathcal{D}}^{\text{row}}_{(n_1,n_2,H)}(l,u) := \quad & \{W \in \mathbb{R}^{n_1\times n_2}_+ \mid \exists(\alpha,\beta,\gamma,z) \in \mathbb{R}^{n_1\times n_2\times H} \times \mathbb{R}^{n_1\times n_2} \times \mathbb{R}^H \times \{0,1\}^{n_2\times H} \mid \\ & (5.34) - (5.37), \\ & l_i \leq \textstyle\sum_{j=1}^{n_2} W_{ij} \leq u_i, \qquad \forall i \in [n_1], \\ & W_{ij} = \textstyle\sum_{h=1}^{H} 2^{-h}\alpha_{ijh} + \beta_{ij}, \qquad \forall i \in [n_1], \forall j \in [n_2]\}. \end{aligned}$$

Proposition 16. *For any $H \in \mathbb{Z}_{++}$, we have $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u) \subseteq \overline{\mathcal{D}}^{\text{row}}_{(n_1,n_2,H)}(l,u)$.*

We will later use the set $\overline{\mathcal{D}}^{\text{row}}_{(n_1,n_2,H)}(l,u)$ to obtain MILP relaxations of the pooling problem.

Now, let us continue with the inner-approximation (or restriction) and discretize the t_j variable as

$$t_j = \frac{2^H}{2^H - 1}\sum_{h=1}^{H} 2^{-h} z_{jh}.$$

This time, equation (5.22) is rewritten as

$$W_{ij} = \left(\sum_{j'=1}^{n_2} W_{ij'}\right)\left(\frac{2^H}{2^H-1}\sum_{h=1}^{H} 2^{-h} z_{jh}\right) \qquad \forall i \in [n_1], \forall j \in [n_2].$$

Finally, we obtain the following inner-approximation of $\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$:

$$\begin{aligned} \underline{\mathcal{D}}^{\text{row}}_{(n_1,n_2,H)}(l,u) := \quad & \{W \in \mathbb{R}^{n_1\times n_2}_+ \mid \exists(\alpha,z) \in \mathbb{R}^{n_1\times n_2\times H} \times \mathbb{R}^{n_1\times n_2} \times \mathbb{R}^H \times \{0,1\}^{n_2\times H} \mid \\ & (5.34) - (5.35), \\ & l_i \leq \textstyle\sum_{j=1}^{n_2} W_{ij} \leq u_i, \qquad \forall i \in [n_1], \\ & W_{ij} = \textstyle\frac{2^H}{2^H-1}\sum_{h=1}^{H} 2^{-h}\alpha_{ijh}, \qquad \forall i \in [n_1], \forall j \in [n_2]\}. \end{aligned}$$

Proposition 17. *For any $H \in \mathbb{Z}_{++}$, we have $\underline{\mathcal{D}}^{\text{row}}_{(n_1,n_2,H)}(l,u) \subseteq \mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)$.*

We will later use the set $\underline{\mathcal{D}}^{\text{row}}_{(n_1,n_2,H)}(l,u)$ to obtain MILP restrictions of the pooling problem.

One can analogously define similar inner and outer-approximations, denoted respectively as $\underline{\mathcal{D}}^{\text{col}}_{(n_1,n_2,H)}(l',u')$ and $\overline{\mathcal{D}}^{\text{col}}_{(n_1,n_2,H)}(l',u')$, for the set $\mathcal{U}^{\text{col}}_{(n_1,n_2)}(l',u')$ as well.

5.3.3 Application of convex hull results to the pooling problem

Pooling problems constitute an important class of non-convex optimization problems in chemical engineering and process design. In this section, we will first review the *multi-commodity flow formulation* of the generalized pooling problem, and then explain how the convex hull results in Section 5.3.1 can be applied to obtain strong relaxations.

Let us first introduce our notation of the generalized pooling problem, which primarily coincides with [6]. Let $G = (N, A)$ be a graph with the node set N and the arc set A. We will denote the set of source (or input), intermediate (or pool) and terminal (or output) nodes as S, I and T, respectively. We have that $N = S \cup I \cup T$ and $A \subseteq (S \times (I \cup T)) \cup (I \times (I \cup T))$.

We will denote the set of source nodes from which there is a path (not necessarily direct) to node i as S_i, and the set of terminal nodes to which there is a path (not necessarily direct) from node i as T_i. We also define

$$N_i^+ := \{j \mid (i,j) \in A\} \text{ and } N_i^- := \{j \mid (j,i) \in A\}.$$

Let K be the set of specifications tracked. Assume that the specification k of source s is given as λ_k^s and the desired specification k of terminal t should be in the interval $[\underline{\mu}_k^t, \overline{\mu}_k^t]$. The capacities of a node $i \in N$ and an arc $(i,j) \in A$ are denoted as U_i and u_{ij} (we will also assume possibly trivial lower bounds for each node i and arc (i,j) as L_i and l_{ij}).

Note that in the standard pooling problem there are no arcs between pools, i.e., from the set $(I \times I)$. While the standard pooling problem is NP-hard to solve [5, 69], there are also some positive results. For instance, a polynomial-time n-approximation algorithm (n is the number of output nodes) is presented in [51], convex hull of special substructure (with one pool node) has been studied [91], and recently some very special cases of the pooling problem has been shown to be polynomially solvable [69, 70, 26, 13]. However, none of these positive results apply for the generalized pooling problem in which the corresponding graph has arcs between pools, i.e., from the set $(I \times I)$. It is well-understood that the generalized pooling problem is more challenging (and realistic) than the standard pooling problem.

In the remainder of this section, we present different ways to formulate the feasible region of the generalized pooling problem *from the rank-1 perspective*. We will assume that the objective function is linear in terms of decision variables and specify its exact expression when we discuss the computational experiments since the objective function changes from instance to instance. Then, in Section 5.3.4, we compare all the relaxations introduced here with the ones known from literature.

Source-based rank formulation

Let us define the following two sets of decision variables: Let f_{ij} be the amount of flow from node i to node j and x_{ij}^s be the amount of flow on arc (i, j) *originated* at the source $s \in S_i$.

We will now describe the constraints of the generalized pooling problem. First, we start with the flow-related constraints:

$$L_i \leq \sum_{j \in N_i^-} f_{ji} \leq U_i \qquad \forall i \in I \cup T \tag{5.38}$$

$$L_i \leq \sum_{j \in N_i^+} f_{ij} \leq U_i \qquad \forall i \in S \tag{5.39}$$

$$l_{ij} \leq f_{ij} \leq u_{ij} \qquad \forall (i,j) \in A \cup \{(s,i) : \; i \in I, \; s \in S_i\} \tag{5.40}$$

$$\sum_{j \in N_i^-} x_{ji}^s = \sum_{j \in N_i^+} x_{ij}^s \qquad \forall i \in I, \forall s \in S_i \tag{5.41}$$

$$\sum_{s \in S_i} x_{ij}^s = f_{ij} \qquad \forall (i,j) \in A \tag{5.42}$$

$$\sum_{j \in N_i^+} x_{ij}^s = f_{si} \qquad \forall i \in I, \forall s \in S_i \tag{5.43}$$

$$x_{ij}^s \geq 0 \qquad \forall (i,j) \in A, \forall s \in S_i. \tag{5.44}$$

Here, constraints (5.38) and (5.39) correspond to node capacity while constraint (5.40) corresponds to the capacity of arc (i,j). We note that there may not exist an arc between each (s,i) pair, in which case we will call the quantity f_{si} a *ghost* flow. Constraint (5.41) is a flow conservation constraint for a node i and source s while constraints (5.42) and (5.43) guarantees that the decomposed flow based on the origin sums up to the actual flow for each arc (i,j).

Next, we give the constraints that guarantee the specification requirements at the terminal nodes:

$$\underline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \leq \sum_{j \in N_t^-} \sum_{s \in S_j} \lambda_k^s x_{jt}^s \leq \overline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \qquad \forall t \in T, \forall k \in K. \tag{5.45}$$

Finally, we present different ways to formulate the non-convex constraints of the pooling problem. A standard approach in the literature, which leads to the well-known $pq-$*Formulation* [136, 6], is to define the proportion variables q_i^s representing the fraction of flow at pool i

originated at source s, and include the following bilinear constraints:

$$x_{ij}^s = q_i^s f_{ij} \qquad \forall (i,j) \in A, \forall s \in S_i. \tag{5.46}$$

Our key observation is to rewrite the bilinear constraints in (5.46) as a set of rank restrictions on a matrix consisting of the decomposed flow variables x_{ij}^s as follows:

$$\operatorname{rank}\left(\left[x_{ij}^s\right]_{(s,j)\in S_i\times N_i^+}\right) = 1 \qquad \forall i \in I. \tag{5.47}$$

The consequences of this realization will be discussed in detail when we construct the polyhedral relaxations of the pooling problem.

Polyhedral relaxations: We will now present two ways to convexify the source-based formulation of the pooling problem. The first relaxation is obtained by the convexification of the rank constraints (5.47) via the column-wise extended formulation, and defined over the following polyhedral set:

$$\begin{aligned}\mathcal{F}_1^S := \Bigg\{ & (f,x) \,|\, (5.40)-(5.45), \\ & \left[x_{ij}^s\right]_{(s,j)} \in \operatorname{conv}\left(\mathcal{U}^{\text{col}+}_{(|S_i|,|N_i^+|)}\left(\left[l_{ij}\right]_j, \left[u_{ij}\right]_j, L_i, U_i\right)\right), \forall i \in I \Bigg\}.\end{aligned}$$

We remark that this relaxation is equivalent to the McCormick relaxation of the pq-Formulation in which bilinear equations (5.46) are convexified via the McCormick envelopes and the following implied constraints are added [2, 136]:

$$\sum_{s\in S_i} q_i^s = 1, \forall i \in I,\ L_i q_i^s \le \sum_{j\in N_i^+} x_{ij}^s \le U_i q_i^s, \forall i \in I, \forall s \in S_i.$$

The second relaxation is a strengthening of the previous one with the addition of the row-wise extended formulation of the rank constraints (5.47), and defined as below:

$$\mathcal{F}_2^S := \left\{ (f,x) \in \mathcal{F}_1^S \mid \begin{bmatrix} x_{ij}^s \end{bmatrix}_{(s,j)} \in \operatorname{conv}\left(\mathcal{U}^{\text{row}+}_{(|S_i|,|N_i^+|)} \left(\begin{bmatrix} l_{si} \end{bmatrix}_s, \begin{bmatrix} u_{si} \end{bmatrix}_s, L_i, U_i \right) \right), \forall i \in I \right\}.$$

We also considered a similar relaxation to $\mathcal{F}_1^S$, in which the row-wise (instead of the column-wise) extended formulation is used. However, we empirically observed that such a relaxation is consistently weaker and hence omitted from further discussion.

Discretization relaxations: We will now present three ways to relax the source-based formulation of the pooling problem to obtain dual bounds using discretization techniques. The first relaxation is obtained by the discretization of the rank constraints (5.47) via the column-wise extended formulation, and defined as follows:

$$\mathcal{M}_1^S(H) := \left\{ (f,x) \in \mathcal{F}_1^S \mid \begin{bmatrix} x_{ij}^s \end{bmatrix}_{(s,j)} \in \operatorname{conv}\left(\overline{\mathcal{D}}^{\text{col}}_{(|S_i|,|N_i^+|,H)} \left(\begin{bmatrix} l_{ij} \end{bmatrix}_j, \begin{bmatrix} u_{ij} \end{bmatrix}_j \right) \right), \forall i \in I \right\}.$$

Here, $H \in \mathbb{Z}_+$ defines the discretization level. The other two relaxations are similarly defined as follows:

$$\mathcal{M}_2^S(H) := \left\{ (f,x) \in \mathcal{F}_2^S \mid \begin{bmatrix} x_{ij}^s \end{bmatrix}_{(s,j)} \in \operatorname{conv}\left(\overline{\mathcal{D}}^{\text{col}}_{(|S_i|,|N_i^+|,H)} \left(\begin{bmatrix} l_{ij} \end{bmatrix}_j, \begin{bmatrix} u_{ij} \end{bmatrix}_j \right) \right), \forall i \in I \right\},$$

$$\mathcal{M}_3^S(H) := \left\{ (f,x) \in \mathcal{F}_2^S \mid \begin{bmatrix} x_{ij}^s \end{bmatrix}_{(s,j)} \in \operatorname{conv}\left(\overline{\mathcal{D}}^{\text{row}}_{(|S_i|,|N_i^+|,H)} \left(\begin{bmatrix} l_{si} \end{bmatrix}_s, \begin{bmatrix} u_{si} \end{bmatrix}_s \right) \right), \forall i \in I \right\}.$$

We do not consider simultaneously discretization with respect to row-wise and column-wise extended formulation, i.e.,

$$\overline{\mathcal{D}}^{\text{col}}_{(|S_i|,|N_i^+|,H)} \left(\begin{bmatrix} l_{ij} \end{bmatrix}_j, \begin{bmatrix} u_{ij} \end{bmatrix}_j \right) \cap \overline{\mathcal{D}}^{\text{row}}_{(|S_i|,|N_i^+|,H)} \left(\begin{bmatrix} l_{si} \end{bmatrix}_s, \begin{bmatrix} u_{si} \end{bmatrix}_s \right),$$

since this becomes a very large formulation.

Following the definition of $\overline{\mathcal{D}}^{\text{col}}$, we conclude that $\mathcal{M}_1^S(H)$ and $\mathcal{M}_2^S(H)$ are obtained from discretizing the ratio variables q_i^s in (5.46) over $\mathcal{F}_1^S$ and $\mathcal{F}_2^S$, respectively. To interpret

$\mathcal{M}_3^S(H)$, we first observe that we can rewrite (5.47) as

$$x_{ij}^s = f_{si} q'_{ij} \qquad \forall (i,j) \in A, \forall s \in S_i, \tag{5.48}$$

where we have introduced the artificial ratio variables q'_{ij}. Following the definition of $\overline{\mathcal{D}}^{\text{row}}$, we conclude that $\mathcal{M}_3^S(H)$ is obtained from discretizing these newly introduced ratio variables q'_{ij} over $\mathcal{F}_2^S$.

Discretization restrictions: We will now present two ways to restrict the source-based formulation of the pooling problem to obtain primal bounds. The first restriction is obtained via the discretization of the rank constraints (5.47), and defined as follows:

$$\begin{aligned}\mathcal{G}_1^S(H) := \Bigg\{ & (f,x) \,|\, (5.40)-(5.45),\\ & \begin{bmatrix} x_{ij}^s \end{bmatrix}_{(s,j)} \in \underline{\mathcal{D}}^{\text{col}}_{(|S_i|,|N_i^+|,H)}\left(\begin{bmatrix} l_{ij} \end{bmatrix}_j, \begin{bmatrix} u_{ij} \end{bmatrix}_j, L_i, U_i \right), \forall i \in I \Bigg\},\end{aligned}$$

where $H \in \mathbb{Z}_+$ defines the discretization level. We remark that this restriction is equivalent to discretizing the q variables in each bilinear equation (5.46) of the pq-Formulation.

The second restriction is obtained in a similar way and given below:

$$\begin{aligned}\mathcal{G}_2^S(H) := \Bigg\{ & (f,x) \,|\, (5.40)-(5.45),\\ & \begin{bmatrix} x_{ij}^s \end{bmatrix}_{(s,j)} \in \underline{\mathcal{D}}^{\text{row}}_{(|S_i|,|N_i^+|,H)}\left(\begin{bmatrix} l_{si} \end{bmatrix}_s, \begin{bmatrix} u_{si} \end{bmatrix}_s, L_i, U_i \right), \forall i \in I \Bigg\}.\end{aligned}$$

Terminal-based rank formulation

Let us define a new set of decision variables x_{ij}^t to denote the amount of flow on arc (i,j) *ended* at the terminal $t \in T_j$.

Firstly, we again describe the flow constraints:

$$L_i \le \sum_{j \in N_i^-} f_{ji} \le U_i \qquad \forall i \in I \cup T \tag{5.49}$$

$$L_i \le \sum_{j \in N_i^+} f_{ij} \le U_i \qquad \forall i \in S \tag{5.50}$$

$$l_{ij} \le f_{ij} \le u_{ij} \qquad \forall (i,j) \in A \cup \{(j,t) :\ j \in I,\ t \in T_j\} \tag{5.51}$$

$$\sum_{i \in N_j^-} x_{ij}^t = \sum_{i \in N_j^+} x_{ji}^t \qquad \forall j \in I, \forall t \in T_j \tag{5.52}$$

$$\sum_{t \in T_j} x_{ij}^t = f_{ij} \qquad \forall (i,j) \in A \tag{5.53}$$

$$\sum_{i \in N_j^-} x_{ij}^t = f_{jt} \qquad \forall j \in I, \forall t \in T_j \tag{5.54}$$

$$x_{ij}^t \ge 0 \qquad \forall (i,j) \in A, \forall t \in T_j. \tag{5.55}$$

Secondly, we give the constraints corresponding to the specification requirements:

$$\underline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \le \sum_{(s,j) \in A: s \in S, t \in T_j} \lambda_k^s x_{sj}^t \le \overline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \qquad \forall t \in T, \forall k \in K. \tag{5.56}$$

To obtain a correct formulation of the pooling problem, the following bilinear equations can be added:

$$x_{ij}^t = q_j^t f_{ij} \qquad \forall (i,j) \in A, \forall t \in T_j. \tag{5.57}$$

This formulation, called the $tp-$*Formulation*, was first proposed in [5] for the standard pooling problem, and then extended to the generalized pooling in [28].

We will again rewrite the bilinear constraints in (5.57) as a set of rank restrictions on a matrix consisting of the decomposed flow variables x_{ij}^t as follows:

$$\operatorname{rank}\left(\left[x_{ij}^t\right]_{(i,t) \in N_j^- \times T_j}\right) = 1 \qquad \forall j \in I. \tag{5.58}$$

Polyhedral relaxations: Similar to the relaxations of the source-based formulation in Section 5.3.3, we define the following two polyhedral relaxations based on the terminal formulation:

$$\mathcal{F}_1^T := \Bigg\{ (f,x) \,|\, (5.51) - (5.56),$$
$$\left[x_{ij}^t \right]_{(i,t)} \in \operatorname{conv}\left(\mathcal{U}^{\text{row}+}_{(|N_j^-|,|T_j|)} \left(\left[l_{ij} \right]_i, \left[u_{ij} \right]_i, L_j, U_j \right) \right), \forall j \in I \Bigg\},$$
$$\mathcal{F}_2^T := \left\{ (f,x) \in \mathcal{F}_1^T \mid \left[x_{ij}^t \right]_{(i,t)} \in \operatorname{conv}\left(\mathcal{U}^{\text{col}+}_{(|N_j^-|,|T_j|)} \left(\left[l_{jt} \right]_t, \left[u_{jt} \right]_t, L_j, U_j \right) \right), \forall j \in I \right\}.$$

We remark that relaxation $\mathcal{F}_1^T$ is equivalent to the McCormick relaxation of the tp-Formulation in which bilinear equations (5.57) are convexified via the McCormick envelopes in addition to the implied constraints $\sum_{t \in T_j} q_j^t = 1, \forall j \in I,\ L_j q_j^t \leq \sum_{i \in N_j^-} x_{ij}^t \leq U_j q_j^t, \forall j \in I, \forall t \in T_j$. The second relaxation $\mathcal{F}_2^T$ is a strengthening of the first one by the addition of the column-wise extended formulation.

Like in the case of the source-based formulation, we also considered a similar relaxation to $\mathcal{F}_1^T$, in which the column-wise (instead of the row-wise) extended formulation is used. However, also in this case, we empirically concluded that such a relaxation is weaker.

Discretization relaxations: Similarly to the relaxation of the source-based formulation in Section 5.3.3, we define the following three relaxations based on the terminal formulation:

$$\mathcal{M}_1^T(H) := \left\{ (f,x) \in \mathcal{F}_1^T \mid \left[x_{ij}^t \right]_{(i,t)} \in \operatorname{conv}\left(\overline{\mathcal{D}}^{\text{row}}_{(|N_j^-|,|T_j|,H)} \left(\left[l_{ij} \right]_i, \left[u_{ij} \right]_i \right) \right), \forall j \in I \right\},$$
$$\mathcal{M}_2^T(H) := \left\{ (f,x) \in \mathcal{F}_2^T \mid \left[x_{ij}^t \right]_{(i,t)} \in \operatorname{conv}\left(\overline{\mathcal{D}}^{\text{row}}_{(|N_j^-|,|T_j|,H)} \left(\left[l_{ij} \right]_i, \left[u_{ij} \right]_i \right) \right), \forall j \in I \right\},$$
$$\mathcal{M}_3^T(H) := \left\{ (f,x) \in \mathcal{F}_2^T \mid \left[x_{ij}^t \right]_{(i,t)} \in \operatorname{conv}\left(\overline{\mathcal{D}}^{\text{col}}_{(|N_j^-|,|T_j|,H)} \left(\left[l_{jt} \right]_t, \left[u_{jt} \right]_t \right) \right), \forall j \in I \right\}.$$

Discretization restrictions: Similarly to the restriction of the source-based formulation in Section 5.3.3, we define the following two restrictions based on the terminal formulation:

$$\begin{aligned}\mathcal{G}_1^T(H) :=\Bigg\{&(f,x)\,|\,(5.51)-(5.56),\\ &\begin{bmatrix}x_{ij}^t\end{bmatrix}_{(i,t)} \in \underline{\mathcal{D}}^{\text{row}}_{(|N_j^-|,|T_j|,H)}\left(\begin{bmatrix}l_{ij}\end{bmatrix}_i, \begin{bmatrix}u_{ij}\end{bmatrix}_i, L_j, U_j\right), \forall j \in I\Bigg\},\\ \mathcal{G}_2^T(H) :=\Bigg\{&(f,x)\,|\,(5.51)-(5.56),\\ &\begin{bmatrix}x_{ij}^t\end{bmatrix}_{(i,t)} \in \underline{\mathcal{D}}^{\text{col}}_{(|N_j^-|,|T_j|,H)}\left(\begin{bmatrix}l_{jt}\end{bmatrix}_t, \begin{bmatrix}u_{jt}\end{bmatrix}_t, L_j, U_j\right), \forall j \in I\Bigg\}.\end{aligned}$$

Source and terminal-based rank formulation

Let us define another set of decision variables x_{ij}^{st} to denote the amount of flow on arc (i,j) *originated* at the source $s \in S_i$ and *ended* at the terminal $t \in T_j$. In this formulation, first appeared in [28], we keep all the flow-related constraints of the source and terminal-based formulations and add the following extra flow constraints:

$$\sum_{j\in N_i^-:s\in S_j} x_{ji}^{st} = \sum_{j\in N_i^+:t\in T_j} x_{ij}^{st} \qquad \forall i \in I, \forall s \in S_i, \forall t \in T_i \tag{5.59}$$

$$\sum_{s\in T_i}\sum_{t\in T_j} x_{ij}^{st} = f_{ij} \qquad \forall (i,j) \in A \tag{5.60}$$

$$\sum_{j\in N_i^+}\sum_{t\in T_j} x_{ij}^{st} = f_{si} \qquad \forall i \in I, \forall s \in S_i \tag{5.61}$$

$$\sum_{i\in N_j^-}\sum_{s\in S_i} x_{ij}^{st} = f_{jt} \qquad \forall j \in I, \forall t \in T_j \tag{5.62}$$

$$x_{ij}^{st} \geq 0 \qquad \forall (i,j) \in A, \forall s \in T_i, \forall t \in T_j. \tag{5.63}$$

The constraints corresponding to the specification requirements in this case are as following:

$$\underline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \leq \sum_{(s,j) \in A: s \in S, t \in T_j} \lambda_k^s x_{sj}^{st} \leq \overline{\mu}_k^t \sum_{j \in N_t^-} f_{jt} \qquad \forall t \in T, \forall k \in K. \tag{5.64}$$

Finally, to have a correct formulation we include the following bilinear equations to the model:

$$x_{ij}^{st} = q_{ij}^{st} f_{ij} \qquad \forall (i,j) \in A, \forall s \in S_i, \forall t \in T_j. \tag{5.65}$$

The next bilinear equations are implied, but their McCormick envelopes can strengthen the relaxation,

$$q_{ij}^{st} = q_i^s q_j^t \qquad \forall (i,j) \in A, \forall s \in S_i, \forall t \in T_j. \tag{5.66}$$

We will again rewrite the bilinear constraints in (5.65) as a set of rank restrictions on a matrix consisting of the decomposed flow variables x_{ij}^{st} as follows:

$$\operatorname{rank}\left(\left[x_{ij}^{st}\right]_{(s,t) \in S_i \times T_j}\right) = 1 \qquad \forall (i,j) \in A \cap (I \times I). \tag{5.67}$$

Polyhedral relaxations: Similar to the relaxations of the source and terminal-based formulation in Section 5.3.3 and 5.3.3, we define the following two polyhedral relaxations based on the source-terminal formulation:

$$\begin{aligned} \mathcal{F}_1^{ST} := \Big\{ (f,x) \mid \; & (5.59) - (5.64), \\ & \left[x_{ij}^{st}\right]_{(s,t)} \in \operatorname{conv}\left(\mathcal{U}_{(|S_i|,|T_j|)}^{\text{row}+}\left(\left[l_{si}\right]_s, \left[u_{si}\right]_s, l_{ij}, u_{ij}\right)\right), \forall (i,j) \in A \Big\} \\ & \cap \mathcal{F}_1^S \cap \mathcal{F}_1^T \cap Mc, \end{aligned}$$

$$\mathcal{F}_2^{ST} := \Bigg\{(f,x) \in \mathcal{F}_1^{ST} \mid \left[x_{ij}^{st}\right]_{(s,t)} \in \operatorname{conv}\left(\mathcal{U}^{\text{col+}}_{(|S_i|,|T_j|)}\left(\left[l_{jt}\right]_t, \left[u_{jt}\right]_t, l_{ij}, u_{ij}\right)\right), \forall (i,j) \in A\Bigg\},$$

where Mc denotes the McCormick relaxation of (5.66), and we use "$\cap$" with slight abuse of notation since the sets do not live in the same variable space.

In our preliminary computations, we found that relaxation $\mathcal{F}_2^{ST}$ is too big and numerically unstable. Thus, we omitted from further discussing this relaxation.

5.3.4 Comparison with previous work

Some of the relaxations presented in the previous section coincide with the well known relaxations of the pooling problem from the literature. Table 5.1 provides a summary as to which of these relaxations are already known and which of them, to the best of our knowledge, are new.

Table 5.1: New relaxations vs. known relaxations for standard and generalized pooling.

Formulation	Standard Pooling	Generalized Pooling
$\mathcal{F}_1^S$	Known [136]	Known [6]
$\mathcal{F}_1^T$	Known [5]	Known [28]
$\mathcal{F}_1^{ST}$	Known [28]	Known [28]
$\mathcal{F}_2^S$	Known* [5]	New
$\mathcal{F}_2^T$	Known* [5]	New

We recall that $\mathcal{F}_1^S$ and $\mathcal{F}_1^T$ respectively correspond to the classical pq- and tp-relaxations for standard pooling. In the generalized pooling setting, $\mathcal{F}_1^T$ and $\mathcal{F}_1^S$ correspond to the McCormick relaxations of the *MCF-J-PQ* and *MCF-I-PQ* formulations in [28]. Moreover, $\mathcal{F}_1^{ST}$ coincides with the McCormick relaxation of the *MCF-(I×J)-PQ* formulation, also introduced in [28].

To the best of our knowledge, the relaxations $\mathcal{F}_2^S$ and $\mathcal{F}_2^T$ are new, at least in the case of generalized pooling. In fact, since we have $\mathcal{F}_1^S \cap \mathcal{F}_1^T = \mathcal{F}_2^S = \mathcal{F}_2^T$ for standard pooling, these two relaxations happen to be equivalent to the stp-relaxation introduced in [5], which is the *intersection* of pq- and tp-relaxations (the asterisk in Table 5.1 signalizes this fact). However, for generalized pooling instances, this equivalence no longer holds. Therefore, $\mathcal{F}_2^S$ and $\mathcal{F}_2^T$ are new formulations in the generalized pooling setting. Moreover, our computational experiments have demonstrated that the intersection of $\mathcal{F}_2^S$ and $\mathcal{F}_2^T$ yields the best bounds on average as shown in Table 5.9, which demonstrates the value of formulating the generalized pooling problems with the rank-1 perspective.

5.3.5 Flow interpretation of the rank-1 conditions

Recall that in the pq-formulation, for a fixed pool $i \in L$ and for any source $s \in S_i$, the ratio variables q_i^s are defined as

$$q_i^s = \frac{x_{ij}^s}{f_{ij}} \ \forall j \in N_i^+ \ \Rightarrow \ x_{ij}^s = q_i^s f_{ij} \ \forall j \in N_i^+.$$

Since q_i^s does not depend on $j \in N_i^+$, these equations say that if, let say, 80% of the total flow on a *particular arc* (i,j) originates at source s, then 80% of the total flow on *any arc* $(i,j),\ j \in N_i^+$, originates at source s (see Figure 5.1). In other words, the pq-formulation enforces flow consistency on the terminal side.

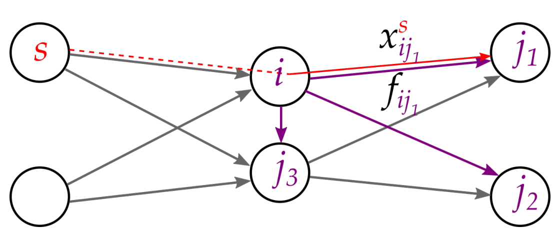

Figure 5.1: Flow consistency enforced by the pq-formulation at pool i with respect to source s.

Similarly, in the tp-formulation, for a fixed pool $j \in L$ and for any terminal $t \in T_j$, the ratio variables q_j^t are defined as

$$q_j^t = \frac{x_{ij}^t}{f_{ij}} \,\forall i \in N_j^- \;\Rightarrow\; x_{ij}^t = q_j^t f_{ij} \,\forall i \in N_j^-.$$

Since q_j^t does not depend on $i \in N_j^-$, these equations say that if, let say, 80% of the total flow on a *particular arc* (i,j) will eventually reach terminal t, then 80% of the total flow on *any arc* $(i,j),\ i \in N_j^-$, will eventually reach terminal t (see Figure 5.2). In other words, the tp-formulation enforces flow consistency on the source side.

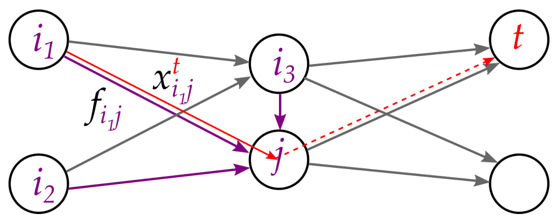

Figure 5.2: Flow consistency enforced by the tp-formulation at pool j with respect to terminal t.

To see how these relations naturally yield the rank-1 condition, we consider again a fixed pool $i \in L$ in the pq-formulation and look at the following matrix:

$$[x_{ij}^s]_{(s,j)} = \begin{bmatrix} x_{i1}^1 & \cdots & x_{ij}^1 & \cdots & x_{in}^1 \\ & & \vdots & & \\ x_{i1}^s & \cdots & x_{ij}^s & \cdots & x_{in}^s \\ & & \vdots & & \\ x_{i1}^m & \cdots & x_{ij}^m & \cdots & x_{in}^m \end{bmatrix}_{(s,j)}.$$

Notice that, for all $s \in S_i$ and $j \in N_i^+$,

$$x_{ij}^s = q_i^s f_{ij}, \;\Leftrightarrow\; [x_{ij}^s]_{(s,j)} = [q_i^s]_s [f_{ij}]_j^\top \;\Leftrightarrow\; \text{rank}([x_{ij}^s]_{(s,j)}) = 1.$$

However, the following also holds for all $s \in S_i$ and $j \in N_i^+$:

$$x_{ij}^s = q_i^s f_{ij}, \Leftrightarrow \; q_i^s = \frac{x_{ij}^s}{\sum_{s'} s_{ij}^{s'}} \; \Leftrightarrow \; \mathrm{rank}([x_{ij}^s]_{(s,j)}) = 1, \tag{5.68}$$

where $f_{ij} = \sum_{s'} x_{ij}^{s'}$. Therefore, for a fixed row s of the matrix $[x_{ij}^s]_{(s,j)}$, the ratio between any element of that row and its column sum does not depend on the column index j. Moreover, as we saw before, by applying our convex hull result to $\mathcal{U}^{\text{col}+}_{(|S_i|,|T_j|)}\left(\Big[l_{jt}\Big]_t, \Big[u_{jt}\Big]_t, l_{ij}, u_{ij}\right)$ for each matrix $[x_{ij}^s]_{(s,j)}$, we obtain the $\mathcal{F}_1^S$ relaxation which coincides with the pq-relaxation.

Now, recall that if a matrix is rank-1, then its transpose is rank-1 as well. Hence, any property that applies to the column must similarly apply to the rows as well. Therefore, in analogy to (5.68), we can define the following for all $s \in S_i$ and $j \in N_i^+$:

$$x_{ij}^s = q'_{ij} f'_{ij}, \Leftrightarrow \; q'_{ij} := \frac{x_{ij}^s}{\sum_{j'} s_{ij'}^{s}} \; \Leftrightarrow \; \mathrm{rank}([x_{ij}^s]_{(s,j)}) = 1, \tag{5.69}$$

where $f'_{ij} := \sum_{j'} s_{ij'}^s$ represents the *total* flow from source s to pool i. Since q'_{ij} does not depend on $s \in S_i$, these equations say that if, let say, 80% of the total flow streaming from a *particular source* s to pool i will eventually reach terminal t, then 80% of the total flow from *any source* $s \in S_i$ will eventually reach terminal t (see Figure 5.3). In other

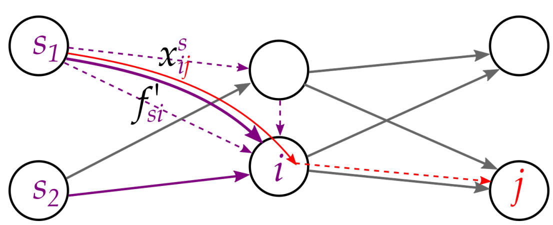

Figure 5.3: Flow consistency enforced by the pq-formulation on the source side via rank-1 condition.

words, the <u>row</u> rank-1 condition on the pq-formulation enforces (partially, for generalized pooling instances) flow consistency on the <u>source</u> side. Thus, by applying our convex hull result to $\mathcal{U}^{\text{row}+}_{(|S_i|,|T_j|)}\left(\Big[l_{jt}\Big]_t, \Big[u_{jt}\Big]_t, l_{ij}, u_{ij}\right)$ for each matrix $[x_{ij}^s]_{(s,j)}$, we partially recover the strength of the tp-formulation, without ever defining the variables of the tp-formulation!

Similarly, by using rank-1 conditions on both row and columns in the tp-formulation, we enforce flow consistency on the source side and partially enforce flow consistency on the terminal side. This explains why $\mathcal{F}_1^S \cap \mathcal{F}_1^T = \mathcal{F}_2^S = \mathcal{F}_2^T$ for standard pooling as we observed in Section 5.3.4.

In conclusion:

- The pq-formulation enforces flow consistency on the terminal side only, and the tp-formulation enforces flow consistency on the source side only;
- The pq-formulation intersected with the tp-formulation yields a stronger formulation;
- A even stronger formulation is given by *MCF-(I×J)-PQ* [28]. It captures the interdependency between the ratio variables q_i^s from the pq-formulation and q_j^t from the tp-formulation. Therefore, it somehow connects flow consistency requirements from both sides of the network;
- Ours "rank-1 pq-formulation" (pq-formulation augmented with the redundant constraints 5.69) enforces the flow consistency requirements on both sides *simultaneously* using much less variables. Same is true for our "rank-1 tp-formulation".
- The rank-1 pq-formulation intersected with the rank-1 tp-formulation yields a stronger formulation, which is incomparable with *MCF-(I×J)-PQ* but is a much smaller formulation (see for instance Table 5.3).

5.4 Computational results

5.4.1 Software and hardware

All of our experiments were run on a Windows 10 machine with 64-bit operating system, x64 based processor with 2.19 GHz, and 32 GB RAM. We used Gurobi 7.5.1 to solve all the linear and mixed-integer linear programs.

5.4.2 Instances

We experimented with three different sets of generalized pooling problem instances. The first set of instances is composed by real-world data from the mining industry [29], the second set is a collection of instances from [6], and the third one was generated by following the recipe proposed in [6].

Mining: We start with a short description of the business problem. Raw material (coal or iron ore, for example) with known quality are blended at points in time in the so-called supply points. Orders for blended product arrive at known points in time and must be satisfied *exactly* by blending previously blended material from the supply points. Costumers specify the minimum quality level for the final blended product. If the raw material is coal, quality level can be determined by the amount of ash and sulfur, for instance. A penalty is incurred if the minimum quality level is violated. The goal is to determine how much material to use from each supply point to meet the demand of each demand while minimizing total penalty. See [29] for more details about the problem, including how it can be formulated as an instance of the generalized pooling problem. We experiment with 24 instances including quarterly, half-yearly and annual planning horizons. Details regarding the size of each instance are displayed in Table B.1.

Literature: In this class, we first considered 40 instances defined in [6]. However, the standard pq-relaxation closes the duality gap for 28 of these instances. Therefore, we only focus on instances L1, L2, L3, L4, L5, L6, L12, L13, L14, L15, C2 and D1 that have non-trivial duality gap, and report the results of the experiments related to these 12 instances. See Table B.2 for further information about them. Instances whose name start with "L", were constructed by the authors of [6] by adding pool-to-pool arcs to standard instances from the literature while instances C2 and D1 were randomly generated. The objective function for all 12 instances is to minimize the total cost associated with each arc.

Random: We generated this set of instances in the same way the authors of [6] generated instances C2 and D1. The only difference is that we increased the number of nodes and changed the density of the network to obtain more challenging instances. In total, we constructed 24 instances as described in Table B.3. The objective function is to minimize the total cost associated with each arc. These instances are available at `https://sites.google.com/view/asteroide-santana/pooling-problem`.

5.4.3 Primal bounds

Primal bounds were available from the literature for both Mining [29] and Literature [6] instances. We compute primal bounds for the Random instances using the MILP discretization restrictions $\mathcal{G}_k^S(H)$ and $\mathcal{G}_k^T(H)$, $k = 1, 2$. In our experiments, we choose the discretization level $H = 3$ for all the instances–meaning that each discretized variable may assume 2^3 different values uniformly distributed within its domain. We then use the best primal bound among these four bounds to compute duality gap. Since the solver can take long to close the MILP duality gap, we set time limit of $1800s$ for each instance. If the MILP is not solved within this time limit, the MILP primal bound is taken as primal bound for the corresponding instance.

Table 5.2 displays the primal bound, the MILP duality gap, and run time, for each discretization restriction. The best performing method is highlighted in bold. Notice that $\mathcal{G}_2^S(H)$, which is not based on the standard pq-formulation, has the best average performance. This is true in terms of finding the best primal bound, closing the MILP duality gap as well as running time.

Table 5.2: Primal bounds via discretization for Random instances. Here, "Bound", "Gap" and "Time" are the MILP primal bound, the percentage optimality gap of the MILP model upon termination, and run time in seconds, respectively.

	$\mathcal{G}_1^S(H)$			$\mathcal{G}_2^S(H)$			$\mathcal{G}_1^T(H)$			$\mathcal{G}_2^T(H)$		
Inst	Bound	Gap	Time	Bound	Gap	Time	Bound	Gap	Time	Bound	Gap	Time
F1	-1933.62	0.00	22.72	**-1968.17**	0.00	5.73	-1968.00	0.01	6.30	-1957.15	0.00	21.05
F2	-4223.78	5.63	1800.02	**-4340.93**	1.52	1800.03	-4265.44	3.69	1800.03	-4287.22	3.56	1800.03
F3	-1293.56	2.96	1800.08	-1344.41	0.00	620.59	**-1344.43**	3.34	1800.06	-1251.69	10.29	1800.08
F4	0.00	0.00	1.30	0.00	0.00	3.27	0.00	0.00	2.20	0.00	0.00	3.11
F5	-2735.68	12.86	1800.02	**-2892.68**	4.41	1800.05	-2721.91	13.40	1800.03	-2685.98	14.75	1800.03
F6	**-4818.56**	8.05	1800.09	-4770.56	8.40	1800.03	-4768.38	9.64	1800.05	-4785.93	8.67	1800.19
F7	-2214.34	35.81	1800.03	**-2431.58**	24.13	1800.03	-2322.47	29.88	1800.05	-2290.95	29.21	1800.06
F8	-2493.85	27.16	1800.17	**-2895.15**	10.06	1800.05	-2893.45	4.73	1800.03	-2733.37	14.15	1800.06
F9	-2290.91	17.91	1800.03	**-2552.66**	1.97	1800.05	-2536.85	4.54	1800.06	-2192.65	21.55	1800.06
F10	-2721.10	5.70	1800.03	**-3006.92**	0.00	1292.48	-2969.85	0.77	1800.05	-2897.09	0.70	1800.03
F11	-2104.35	0.00	26.83	**-2159.43**	0.00	6.55	-2159.43	0.00	7.03	-2143.44	0.00	8.50
F12	-4541.65	2.05	1800.02	**-4594.01**	0.01	1107.47	-4583.04	1.11	1800.06	-4546.62	2.03	1800.05
F13	-2786.81	6.54	1800.03	**-2872.81**	0.01	1320.28	-2867.10	2.22	1800.03	-2812.18	6.09	1800.05
F14	-492.11	0.00	11.27	**-497.23**	0.00	0.91	-497.23	0.00	0.95	-495.81	0.00	4.14
F15	-2365.46	8.43	1800.02	**-2414.56**	10.43	1800.03	-2396.89	11.97	1800.06	-2379.11	13.12	1800.02
F16	-5824.86	0.01	271.45	-5834.27	0.00	274.00	**-5838.04**	0.00	59.13	-5831.01	0.01	326.98
F17	-2100.98	0.00	17.16	-2098.36	0.00	4.41	-2098.36	0.00	8.25	**-2105.35**	0.00	9.37
F18	-744.71	0.01	194.59	**-760.57**	0.00	22.61	-760.55	0.00	14.98	-745.66	0.01	130.30
F19	-555.37	0.00	1.03	**-582.87**	0.00	0.53	**-582.87**	0.01	0.36	-555.37	0.00	0.66
F20	-3419.62	5.44	1800.02	**-3599.41**	0.00	115.48	-3548.90	0.00	318.98	-3537.36	0.01	842.69
F21	-8094.23	30.97	1800.08	**-8417.90**	25.51	1800.09	-7567.30	40.54	1800.14	-7312.21	44.58	1800.05
F22	-3629.75	14.18	1800.00	**-3893.33**	9.51	1800.03	-3854.77	10.53	1800.03	-3828.98	6.23	1800.03
F23	-6611.50	45.11	1800.06	**-6667.33**	43.43	1800.08	-6026.67	61.57	1800.09	-6458.53	49.26	1800.08
F24	-999.81	0.00	140.50	**-1025.35**	0.00	26.97	-1009.19	0.00	51.94	-1005.53	0.00	186.87
Ave.	-2874.86	9.53	1153.65	**-2984.19**	5.81	950.07	-2899.21	8.25	1069.62	-2868.30	9.34	1113.94

5.4.4 Dual bounds via LP relaxations

We compute dual bounds using three types of LP relaxations which we will refer to as "Light", "Medium" and "Heavy". Light LP relaxations are given by $\mathcal{F}_k^S, \mathcal{F}_k^T$, $k = 1, 2$ while Medium LP relaxations are defined as $\mathcal{F}_k^S \cap \mathcal{F}_l^T$, $k, l = 1, 2$. The only Heavy LP relaxation considered is $\mathcal{F}_1^{ST}$ from [28]. Duality gap and run time of each of these nine methods are reported for all the three sets of instances in Tables 5.3, 5.4 and 5.5. The best performing method is highlighted in bold for each instance.

By construction, the method based on $\mathcal{F}_2^S \cap \mathcal{F}_2^T$ gives the strongest dual bound among Light and Medium LP relaxations. It is interesting to observe that it also performs better than the Heavy LP relaxation $\mathcal{F}_1^{ST}$ in terms of both average duality gap and run time. In fact, there are only three instances in which $\mathcal{F}_1^{ST}$ is strictly better than all of the Light and Medium LP relaxations (these are instances 2009Q4, 2011H1 and 2011H2 from the Mining set). On the other hand, the Heavy LP relaxation $\mathcal{F}_1^{ST}$ takes significantly longer than lighter LP relaxations and runs into memory issues occasionally. For these instances, we report the best bound (and respective time) among all the other methods (see numbers in parentheses in Tables 5.3 and 5.5), and use those figures in the averages.

Finally, we point out the relative success of the Light LP relaxations, which typically take much shorter amount of time than their Medium LP counterparts with similar optimality gaps proven. This has motivated us to only focus on the MILP relaxations of the Light LP relaxations as discussed in the next section. We also note that a comparison of the best performing relaxations from the Light and Medium LP relaxations in terms of their average performances is provided later in Table 5.9.

5.4.5 Dual bounds via MILP relaxations

Besides using the LP relaxations, we also compute dual bounds for all the instances using the MILP discretization relaxations $\mathcal{M}_k^S(H)$ and $\mathcal{M}_k^T(H)$, $k = 1, 2, 3$. Recall that $\mathcal{M}_1^S(H)$ and $\mathcal{M}_2^S(H)$ are obtained from discretizing the ratio variables q_j^s in (5.46), whereas $\mathcal{M}_3^S(H)$ is obtained from discretizing the newly introduced ratio variables q_{ij} in (5.48). Similar interpretation holds for the terminal-based formulations.

In our experiments, we choose the discretization level $H = 3$ for all the instances– meaning that the domain of each discretized variable is partitioned into 2^3 intervals of equal

Table 5.3: Duality gaps for Mining instances via LP. Here, "Paper" is the relaxation used in [29], the paper that introduced these instances in the literature.

	Paper [29]		$\mathcal{F}_1^S$		$\mathcal{F}_2^S$		$\mathcal{F}_1^T$		$\mathcal{F}_2^T$		$\mathcal{F}_1^S \cap \mathcal{F}_1^T$		$\mathcal{F}_2^S \cap \mathcal{F}_1^T$		$\mathcal{F}_1^S \cap \mathcal{F}_2^T$		$\mathcal{F}_2^S \cap \mathcal{F}_2^T$		$\mathcal{F}_1^{ST}$	
Inst	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
2009H2	37.00	0.05	7.13	0.53	4.06	1.12	4.58	1.23	4.58	9.28	4.27	5.13	**3.59**	5.89	4.27	7.16	**3.59**	6.64	4.02	1697.27
2009Q3	29.36	0.03	3.84	0.06	1.59	0.23	2.16	0.08	2.16	0.17	2.01	0.34	**1.45**	0.69	2.01	0.61	**1.45**	0.47	1.96	1.23
2009Q4	41.86	0.02	24.93	0.13	14.49	0.57	17.11	0.30	17.11	0.78	16.94	1.20	13.77	1.75	16.94	1.36	13.77	1.98	**10.84**	22.59
2010Y	26.01	0.25	10.37	5.16	8.58	10.99	8.91	33.62	8.91	61.63	8.87	101.08	**8.14**	86.91	8.87	141.66	**8.14**	92.30	**(8.14)**	86.91
2010H1	32.42	0.08	14.64	1.02	13.42	1.95	13.38	8.97	13.38	6.23	13.32	10.09	**12.77**	17.47	13.32	18.48	**12.77**	15.73	12.78	285.80
2010H2	14.83	0.09	3.94	0.81	1.84	2.09	2.45	3.27	2.45	5.34	2.45	10.60	**1.80**	16.55	2.45	15.61	**1.80**	14.89	2.26	480.52
2010Q1	19.74	0.03	4.35	0.20	3.90	0.75	4.31	0.55	4.31	0.66	4.14	2.03	**3.88**	1.81	4.14	2.50	**3.88**	2.36	3.94	13.33
2010Q2	35.38	0.03	21.41	0.11	18.59	0.61	18.58	0.27	18.58	0.61	18.55	1.28	**17.90**	1.39	18.55	1.38	**17.90**	0.98	18.26	5.66
2010Q3	20.33	0.02	3.84	0.09	1.54	4.05	2.56	0.13	2.56	0.39	2.56	0.61	**1.49**	0.94	2.56	1.11	**1.49**	0.98	2.23	4.20
2010Q4	28.62	0.05	11.43	0.12	9.95	0.36	9.43	0.58	9.43	0.94	9.14	1.00	**8.83**	1.37	9.14	1.48	**8.83**	1.56	8.75	47.97
2011Y	19.30	0.14	2.45	2.83	1.50	7.64	1.43	7.95	1.43	16.39	1.41	57.67	**1.25**	322.28	1.41	60.89	**1.25**	88.55	1.28	1161.44
2011H1	9.07	0.06	2.30	0.48	1.40	0.91	1.24	0.83	1.24	1.91	1.23	2.36	1.13	1.95	1.23	2.77	1.13	5.39	**1.10**	14.53
2011H2	22.21	0.06	2.17	0.77	1.43	1.56	1.37	1.53	1.37	2.38	1.34	7.75	1.26	10.12	1.34	12.62	1.26	21.31	**1.24**	172.06
2011Q1	10.51	0.02	1.78	0.04	1.03	0.15	0.83	0.11	0.83	0.34	0.82	0.52	**0.72**	0.53	0.82	0.47	**0.72**	0.77	**0.72**	1.16
2011Q2	4.04	0.02	3.19	0.05	2.29	0.16	2.53	0.12	2.53	0.40	2.53	0.34	**2.10**	0.45	2.53	0.63	**2.10**	0.98	2.24	3.75
2011Q3	10.04	0.00	0.39	0.02	0.07	0.03	0.15	0.03	0.15	0.06	0.15	0.14	**0.07**	0.08	0.15	0.11	**0.07**	0.14	0.15	0.25
2011Q4	16.09	0.02	1.01	0.06	0.57	0.10	0.99	0.09	0.99	0.30	0.90	0.53	**0.56**	0.73	0.90	0.63	**0.56**	0.97	0.83	3.08
2012Y	8.20	0.11	2.79	1.78	1.72	5.61	2.37	7.56	2.37	16.38	2.19	21.64	**1.57**	30.39	2.19	29.50	**1.57**	41.34	**(1.57)**	30.39
2012H1	4.99	0.08	3.35	1.27	2.20	2.50	3.00	2.34	3.00	5.17	2.77	10.77	**2.02**	16.66	2.77	13.19	**2.02**	28.83	**(2.02)**	16.66
2012H2	10.21	0.02	0.97	0.11	0.42	0.47	0.74	0.19	0.74	0.33	0.66	1.02	**0.42**	0.72	0.66	0.91	**0.42**	1.33	0.64	4.75
2012Q1	13.93	0.01	9.31	0.03	4.43	0.08	8.66	0.09	8.66	0.22	8.25	0.41	**4.16**	0.33	8.25	0.31	**4.16**	0.75	8.13	2.37
2012Q2	1.68	0.02	0.70	0.03	0.30	0.09	0.49	0.08	0.49	0.23	0.49	0.34	**0.21**	0.28	0.49	0.53	**0.21**	0.41	0.49	0.95
2012Q3	5.92	0.02	1.13	0.05	0.56	0.13	0.99	0.06	0.99	0.20	0.92	0.38	**0.56**	0.33	0.92	0.42	**0.56**	0.86	0.89	1.52
2012Q4	25.81	0.00	4.91	0.02	3.50	0.02	1.91	0.03	1.91	0.03	1.91	0.05	**1.91**	0.06	1.91	0.08	**1.91**	0.09	1.91	0.22
Ave.	18.65	0.05	5.93	0.66	4.14	1.76	4.59	2.92	4.59	5.43	4.49	9.89	**3.82**	21.65	4.49	13.10	**3.82**	13.73	4.02	169.11

Table 5.4: Duality gaps for Literature instances via LP.

	$\mathcal{F}_1^S$		$\mathcal{F}_2^S$		$\mathcal{F}_1^T$		$\mathcal{F}_2^T$		$\mathcal{F}_1^S \cap \mathcal{F}_1^T$		$\mathcal{F}_2^S \cap \mathcal{F}_1^T$		$\mathcal{F}_1^S \cap \mathcal{F}_2^T$		$\mathcal{F}_2^S \cap \mathcal{F}_2^T$		$\mathcal{F}_1^{ST}$	
Inst	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
L1	1.01	0.00	**0.86**	0.00	1.01	0.00	1.01	0.00	1.01	0.02	**0.86**	0.00	1.01	0.00	**0.86**	0.00	1.01	0.00
L2	55.24	0.00	55.24	0.02	55.74	0.00	**52.83**	0.00	55.24	0.00	55.24	0.02	**52.83**	0.02	**52.83**	0.00	55.24	0.02
L3	**4.55**	0.00	**4.55**	0.00	**4.55**	0.00	**4.55**	0.00	**4.55**	0.00	**4.55**	0.02	**4.55**	0.00	**4.55**	0.00	**4.55**	0.03
L4	**2.45**	0.02	**2.45**	0.06	**2.45**	0.00	**2.45**	0.00	**2.45**	0.02	**2.45**	0.05	**2.45**	0.03	**2.45**	0.09	**2.45**	0.16
L5	10.80	0.02	10.80	0.00	11.26	0.00	**9.60**	0.02	10.80	0.02	10.80	0.02	**9.60**	0.02	**9.60**	0.03	10.80	0.03
L6	22.22	0.00	22.06	0.00	**20.37**	0.00	**20.37**	0.00	**20.37**	0.00	**20.37**	0.00	**20.37**	0.00	**20.37**	0.00	**20.37**	0.00
L12	**25.00**	0.02	**25.00**	0.00	**25.00**	0.00	**25.00**	0.00	**25.00**	0.00	**25.00**	0.00	**25.00**	0.00	**25.00**	0.00	**25.00**	0.00
L13	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00	**66.67**	0.00
L14	16.67	0.00	16.67	0.00	16.67	0.00	**6.67**	0.00	16.67	0.00	16.67	0.00	**6.67**	0.00	**6.67**	0.00	16.67	0.00
L15	37.41	0.00	**25.88**	0.00	**25.88**	0.00	**25.88**	0.00	**25.88**	0.02	**25.88**	0.00	**25.88**	0.00	**25.88**	0.02	**25.88**	0.00
C2	**1.23**	0.03	**1.23**	0.02	**1.23**	0.03	**1.23**	0.03	**1.23**	0.08	**1.23**	0.08	**1.23**	0.09	**1.23**	0.06	**1.23**	0.33
D1	**1.06**	0.09	**1.06**	0.25	**1.06**	0.12	**1.06**	0.22	**1.06**	0.55	**1.06**	0.94	**1.06**	0.94	**1.06**	1.17	**1.06**	3.92
Ave.	20.36	0.01	19.37	0.03	19.32	0.01	18.11	0.02	19.24	0.06	19.23	0.09	18.11	0.09	**18.10**	0.11	19.24	0.37

length. Since the solver can take long to close the MILP duality gap, we set time limit of $1800s$ for each instance. If the MILP is not solved within this time limit, the MILP dual bound is taken as dual bound for the corresponding instance. The results for all instances are reported in Tables 5.6, 5.7 and 5.8. The best performing method is highlighted in bold.

For the Mining instances, $\mathcal{M}_3^S(H)$ is the best performing method for 19 out of the 24 instances (see Table 5.6). The average run time of $\mathcal{M}_3^S(H)$ is also one of the best. For the Literature instances, the terminal-based formulation works better. As we can see in Table 5.7, on average, $\mathcal{M}_3^T(H)$ closes almost twice more gap than the second best performing method. For the Random instances, $\mathcal{M}_3^S(H)$ is again the best performing method. On average, $\mathcal{M}_3^S(H)$ yields the best gap and the best run time (see Table 5.8). However, there are

Table 5.5: Duality gaps for Random instances via LP.

	$\mathcal{F}_1^S$		$\mathcal{F}_2^S$		$\mathcal{F}_1^T$		$\mathcal{F}_2^T$		$\mathcal{F}_1^S \cap \mathcal{F}_1^T$		$\mathcal{F}_2^S \cap \mathcal{F}_1^T$		$\mathcal{F}_1^S \cap \mathcal{F}_2^T$		$\mathcal{F}_2^S \cap \mathcal{F}_2^T$		$\mathcal{F}_1^{ST}$	
Inst	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
F1	**3.79**	0.02	**3.79**	0.02	5.02	0.03	4.09	0.06	**3.79**	0.05	**3.79**	0.12	**3.79**	0.09	**3.79**	0.09	**3.79**	0.22
F2	4.81	0.39	4.60	1.02	5.15	0.41	4.50	0.89	4.30	1.27	4.29	3.43	**4.28**	3.06	**4.28**	5.98	4.29	13.70
F3	13.52	0.27	**13.19**	1.27	14.57	0.70	14.57	1.41	**13.19**	1.55	**13.19**	2.82	**13.19**	2.00	**13.19**	6.44	13.19	22.39
F4	**0.00**	0.14	**0.00**	2.48	**0.00**	1.03	**0.00**	1.72	**0.00**	1.31	**0.00**	4.80	**0.00**	3.37	**0.00**	6.25	**0.00**	6.61
F5	14.58	0.42	11.02	1.81	13.06	0.34	11.96	1.06	11.71	1.19	**10.97**	3.59	11.66	3.03	**10.97**	6.19	11.65	22.69
F6	10.69	0.31	10.03	1.73	12.25	0.27	10.45	1.17	10.15	3.19	9.92	8.11	9.97	6.56	**9.91**	12.42	9.94	60.92
F7	33.72	1.05	33.09	3.25	34.91	1.20	34.04	5.08	33.09	5.31	33.09	12.50	**33.05**	12.45	**33.05**	22.45	33.09	86.13
F8	17.15	0.58	**16.90**	1.77	17.46	0.53	17.06	1.81	**16.90**	4.22	**16.90**	8.47	**16.90**	7.89	**16.90**	14.05	**16.90**	22.58
F9	18.13	0.50	17.17	1.03	20.75	0.38	17.78	1.15	17.24	3.48	**17.15**	8.91	17.16	5.77	**17.15**	16.30	17.16	19.64
F10	7.84	0.31	**7.81**	1.30	8.08	0.61	7.86	1.41	**7.81**	2.66	**7.81**	7.27	**7.81**	4.25	**7.81**	9.17	**7.81**	12.98
F11	5.43	0.02	5.32	0.05	5.71	0.03	5.20	0.08	5.28	0.06	5.28	0.16	**5.19**	0.16	**5.19**	0.23	5.28	0.39
F12	4.57	0.14	4.54	0.30	5.14	0.19	4.78	0.34	4.48	1.06	4.48	1.69	**4.47**	1.64	**4.47**	2.31	4.48	8.73
F13	7.20	0.67	7.15	2.11	8.12	0.69	7.92	1.81	7.17	1.86	7.14	4.45	7.16	3.84	**7.13**	7.48	7.16	32.11
F14	**0.00**	0.08	**0.00**	0.48	**0.00**	0.30	**0.00**	0.80	**0.00**	0.95	**0.00**	2.00	**0.00**	2.02	**0.00**	1.75	**0.00**	3.83
F15	17.95	0.23	17.88	0.78	18.82	0.39	18.33	1.61	16.36	1.84	**16.33**	2.77	16.36	3.03	**16.33**	6.06	16.36	14.53
F16	4.64	0.06	4.29	0.09	4.41	0.05	4.31	0.11	4.28	0.28	**4.26**	0.42	4.28	0.38	**4.26**	0.64	4.27	2.31
F17	10.24	0.02	10.24	0.05	9.63	0.06	**9.62**	0.11	**9.62**	0.12	**9.62**	0.16	**9.62**	0.17	**9.62**	0.27	**9.62**	0.64
F18	**16.63**	0.09	**16.63**	0.34	**16.63**	0.19	**16.63**	0.56	**16.63**	0.80	**16.63**	0.73	**16.63**	1.08	**16.63**	2.53	**16.63**	1.87
F19	**1.06**	0.02	**1.06**	0.02	2.73	0.03	**1.06**	0.03	**1.06**	0.03	**1.06**	0.03	**1.06**	0.05	**1.06**	0.09	**1.06**	0.08
F20	12.74	0.11	12.11	0.16	13.38	0.14	12.28	0.31	11.52	0.91	11.48	0.88	11.50	1.00	**11.46**	1.34	11.52	2.63
F21	26.02	12.38	25.57	55.88	26.49	8.72	25.71	50.89	25.42	70.72	25.40	165.03	25.37	133.67	**25.36**	310.98	(**25.36**)	310.98
F22	22.64	0.23	22.17	0.55	23.43	0.34	22.61	0.66	21.91	1.42	**21.91**	3.19	**21.91**	2.86	**21.91**	4.34	**21.91**	10.83
F23	43.92	10.64	43.58	49.09	47.04	12.15	45.05	49.55	43.39	53.61	43.35	493.14	43.16	224.12	**43.13**	207.30	(**43.13**)	207.30
F24	11.19	0.22	11.19	0.50	11.19	1.02	**11.18**	2.23	11.19	2.23	11.19	5.81	**11.18**	7.45	**11.18**	11.38	11.19	5.44
Ave.	12.85	1.20	12.47	5.25	13.50	1.24	12.79	5.20	12.35	6.67	12.30	30.85	12.32	17.91	**12.28**	27.34	12.32	36.23

instances in which $\mathcal{M}_3^T(H)$ performs significantly better, for example, instances F10 and F11. Thus, it is difficult to advise a single method in this case.

In Table 5.9, we compare the performances of the average best LP and MILP methods on each instance set. As expected, the average run time of each MILP method was much higher than its LP counterpart. On the other hand, MILP methods can close significantly more gap than the LP ones. The performance discrepancy is more evident in the Literature instances due to their relatively small sizes. Another interesting observation is that the winning Light LP method seems to suggest which MILP method will perform better. For instance, the source-based MILP relaxations perform better for the Mining and Random instances, as correctly predicted by the better performance of the source-based light LP relaxations. This situation is reversed for the Literature instances as the terminal-based light LP and MILP relaxations seem to provide stronger relaxations consistently.

5.5 Conclusion

We propose new convex relaxations for QCQPs derived from its rank-based formulation (5.2)–(5.5). Specifically, we study the convex hull of sets defined by a rank-1 constraint

Table 5.6: Duality gaps via discretization for Mining instances.

	$\mathcal{M}_1^S(H)$		$\mathcal{M}_2^S(H)$		$\mathcal{M}_3^S(H)$		$\mathcal{M}_1^T(H)$		$\mathcal{M}_2^T(H)$		$\mathcal{M}_3^T(H)$	
Inst	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
2009H2	3.64	1800.17	2.69	1800.18	**1.61**	1800.09	2.08	1800.37	2.08	1800.27	2.65	1800.14
2009Q3	0.80	23.67	0.54	16.45	**0.20**	10.64	0.21	12.56	0.22	18.12	0.53	8.56
2009Q4	8.82	1800.05	8.41	1800.05	**3.45**	1558.98	4.93	1800.06	4.74	1800.06	4.69	1800.02
2010Y	9.13	1800.27	8.27	1801.63	**8.09**	1800.63	8.06	1800.33	8.24	1800.25	8.44	1800.14
2010H1	13.31	1800.07	11.71	1800.15	**6.92**	1800.23	9.05	1800.09	8.55	1800.23	8.41	1800.09
2010H2	1.22	1800.11	**0.84**	1800.11	1.03	1800.33	1.00	1800.19	0.93	1800.11	2.19	1800.05
2010Q1	2.89	232.87	2.75	379.08	**1.64**	134.11	2.25	665.41	2.25	1279.26	2.32	199.84
2010Q2	12.37	1028.67	11.64	1594.03	**2.95**	232.52	4.89	1800.03	4.85	1800.05	4.23	422.70
2010Q3	0.94	39.75	0.70	40.09	**0.38**	14.31	0.70	33.23	0.70	37.08	0.80	48.44
2010Q4	6.38	132.55	5.28	160.81	**1.97**	772.50	3.84	231.61	3.84	239.50	4.95	701.95
2011Y	1.56	1800.09	1.28	1800.19	1.20	1800.11	1.17	1800.16	**1.13**	1800.20	1.28	1800.84
2011H1	0.64	1800.09	0.54	1800.08	**0.18**	871.16	0.41	1800.05	0.58	1800.14	0.31	287.62
2011H2	1.37	1800.06	1.26	1800.09	**0.37**	1800.09	1.07	1800.09	1.05	1800.12	0.66	1800.05
2011Q1	0.31	18.62	0.26	36.94	**0.12**	57.71	0.22	20.05	0.22	24.11	0.22	29.84
2011Q2	1.08	90.28	1.01	38.64	**0.49**	36.70	0.89	62.84	0.89	111.09	0.69	58.31
2011Q3	0.05	0.66	0.03	0.77	**0.01**	1.13	0.04	1.33	0.04	1.81	0.06	0.83
2011Q4	0.18	26.39	0.15	31.48	**0.10**	8.00	0.24	45.20	0.24	39.56	0.17	49.69
2012Y	1.85	1800.06	1.43	1800.09	**1.20**	1802.70	1.37	1800.08	1.37	1800.06	1.95	1800.13
2012H1	2.22	1800.03	1.72	1800.08	1.34	1800.04	**1.14**	1800.06	1.39	1800.08	2.48	1800.11
2012H2	0.19	39.77	0.15	52.55	**0.07**	52.66	0.18	55.59	0.18	60.58	0.19	72.69
2012Q1	3.35	22.00	2.15	10.64	**0.91**	11.08	1.32	9.00	1.32	23.47	1.78	23.52
2012Q2	0.16	12.02	0.12	10.39	0.12	3.97	0.16	18.55	0.16	19.77	**0.11**	22.23
2012Q3	0.29	19.80	0.23	28.28	**0.07**	28.56	0.18	20.72	0.18	48.72	0.21	33.38
2012Q4	1.77	4.73	1.65	4.16	0.70	1.03	0.44	2.31	0.44	2.22	**0.33**	8.56
Ave.	3.10	820.53	2.70	850.29	**1.46**	758.30	1.91	874.16	1.90	904.45	2.07	757.07

Table 5.7: Duality gaps via discretization for Literature instances.

	$\mathcal{M}_1^S(H)$		$\mathcal{M}_2^S(H)$		$\mathcal{M}_3^S(H)$		$\mathcal{M}_1^T(H)$		$\mathcal{M}_2^T(H)$		$\mathcal{M}_3^T(H)$	
Inst	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
L1	0.33	0.03	0.33	0.05	**0.26**	0.05	0.36	0.05	0.36	0.05	0.59	0.03
L2	2.96	1.12	2.96	1.70	15.66	1.38	2.70	0.20	2.70	0.09	**1.55**	0.28
L3	2.96	0.50	2.96	0.86	3.58	0.37	1.97	0.08	1.97	0.09	**1.55**	0.17
L4	1.96	16.64	1.96	19.78	2.45	1.36	0.47	0.49	0.47	0.28	**0.26**	2.14
L5	2.88	1.17	2.88	1.58	4.19	12.78	0.96	0.81	0.96	0.68	**0.05**	0.14
L6	**0.00**	0.14	**0.00**	0.17	**0.00**	0.11	**0.00**	0.04	**0.00**	0.02	**0.00**	0.06
L12	**0.00**	0.09	**0.00**	0.11	**0.00**	0.12	**0.00**	0.02	**0.00**	0.02	**0.00**	0.03
L13	**0.00**	0.08	**0.00**	0.13	**0.00**	0.20	**0.00**	0.02	**0.00**	0.03	**0.00**	0.03
L14	**0.00**	0.10	**0.00**	0.11	4.76	0.17	1.91	0.02	1.91	0.04	**0.00**	0.03
L15	0.77	0.14	**0.68**	0.12	1.20	0.08	1.20	0.03	1.20	0.06	**0.68**	0.09
C2	0.14	0.64	0.14	0.72	**0.10**	0.59	**0.10**	0.55	**0.10**	0.69	0.14	0.47
D1	1.06	1800.03	1.06	1800.00	1.04	1800.02	**1.02**	1800.02	1.03	1800.03	1.04	1800.02
Ave.	1.09	151.72	1.08	152.11	2.77	151.44	0.89	150.19	0.89	150.17	**0.49**	150.29

intersected with some linear side constraints (5.6). For several choices of linear side constraints, we show that this convex hull is polyhedral or SOCr, and provide compact formulations for the polyhedral cases. We also show that in all these cases, a linear objective can be optimized over these sets in polynomial time.

Table 5.8: Duality gaps via discretization for Random instances.

	$\mathcal{M}_1^S(H)$		$\mathcal{M}_2^S(H)$		$\mathcal{M}_3^S(H)$		$\mathcal{M}_1^T(H)$		$\mathcal{M}_2^T(H)$		$\mathcal{M}_3^T(H)$	
Inst	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
F1	**2.04**	7.06	**2.04**	9.52	2.30	4.14	2.30	6.31	2.30	4.13	2.26	10.23
F2	3.01	1800.03	2.99	1800.02	**2.39**	1800.08	2.63	1800.08	2.59	1800.08	2.63	1800.03
F3	3.42	1800.05	4.36	1800.05	**2.01**	526.69	7.50	1800.05	8.88	1800.03	5.42	1800.09
F4	**0.00**	2.44	**0.00**	3.19	**0.00**	3.91	**0.00**	3.20	**0.00**	3.78	**0.00**	4.02
F5	7.26	1800.06	7.90	1800.03	6.52	1800.03	6.78	1800.06	7.55	1800.05	**6.27**	1800.06
F6	8.25	1800.02	8.38	1800.03	**8.12**	1800.05	9.11	1800.05	8.29	1800.05	8.19	1800.06
F7	24.44	1800.05	25.67	1800.06	25.96	1800.05	24.79	1800.06	26.83	1800.04	**22.87**	1800.08
F8	10.40	1800.06	10.81	1800.05	9.53	1800.05	**7.66**	1800.08	8.91	1800.06	10.55	1800.03
F9	8.09	1800.06	8.29	1800.06	6.78	1800.05	8.25	1800.06	7.88	1800.06	**6.37**	1800.12
F10	3.08	1800.14	3.65	1800.14	4.23	726.17	4.69	764.77	4.69	653.10	**2.47**	1110.62
F11	1.25	23.02	1.25	26.14	1.30	63.05	1.47	57.05	1.47	71.55	**0.84**	9.50
F12	1.75	1800.05	1.62	1800.07	**1.11**	431.17	1.57	1800.14	1.61	1800.12	1.49	1800.16
F13	4.54	1800.05	4.22	1800.05	**1.67**	1751.19	3.81	1800.10	3.79	1800.06	4.40	1800.05
F14	**0.00**	1.09	**0.00**	0.69	**0.00**	0.88	**0.00**	1.33	**0.00**	2.36	**0.00**	3.08
F15	13.15	1800.03	12.73	1800.03	**12.22**	1800.03	12.97	1800.03	13.52	1800.05	12.43	1800.05
F16	1.37	240.83	1.35	347.73	1.16	161.31	1.15	573.34	**1.09**	1460.55	1.48	39.92
F17	1.94	10.59	1.94	19.34	**1.82**	6.59	1.83	6.95	1.83	10.22	2.50	10.50
F18	1.79	61.87	1.79	42.75	**1.16**	36.03	1.17	21.73	1.17	27.42	2.64	16.83
F19	**0.00**	0.75	**0.00**	0.56	**0.00**	0.38	**0.00**	0.39	**0.00**	0.39	**0.00**	0.44
F20	3.47	1800.11	3.50	1800.05	2.49	535.50	2.29	631.09	2.29	1337.64	**2.23**	308.12
F21	26.01	1800.05	**25.46**	1800.08	25.55	1800.06	26.20	1800.05	25.69	1800.09	25.46	1800.11
F22	10.40	1800.03	11.68	1800.03	11.11	1800.05	12.60	1800.04	12.95	1800.08	7.26	1800.13
F23	43.65	1800.06	**43.58**	1800.11	**43.58**	1800.09	46.52	1800.05	45.03	1800.09	44.69	1800.09
F24	5.96	46.45	5.96	39.28	**5.88**	12.58	5.99	12.37	5.99	28.78	5.92	20.03
Ave.	7.72	1141.46	7.88	1145.42	**7.37**	927.50	7.97	1061.64	8.10	1125.03	7.43	1038.93

Table 5.9: Best average duality gap for each set of instances. Here, "Method" is the method that yields the best average duality gap.

	Light LP			LP			MILP		
Instance Set	Gap	Time	Method	Gap	Time	Method	Gap	Time	Method
Mining	4.14	1.76	$\mathcal{F}_2^S$	3.82	13.73	$\mathcal{F}_2^S \cap \mathcal{F}_2^T$	1.46	758.30	$\mathcal{M}_3^S(H)$
Literature	18.11	0.02	$\mathcal{F}_2^T$	18.10	0.11	$\mathcal{F}_2^S \cap \mathcal{F}_2^T$	0.49	150.29	$\mathcal{M}_3^T(H)$
Random	12.47	5.25	$\mathcal{F}_2^S$	12.28	27.34	$\mathcal{F}_2^S \cap \mathcal{F}_2^T$	7.37	927.50	$\mathcal{M}_3^S(H)$

On the application side, we propose rank-1 based formulations for the pooling problem. The new formulations combined with our convexification results allow us to derive new convex relaxations for the pooling problem, which we show to generalize, for example, the well-known pq-relaxation. Studying the pooling problem via rank-based formulations not only allows us to recover previous relaxations from the literature, but also leads us to improve/strengthen them in a systematic way. In addition, inspired by our newly proposed formulation and our convexification results, we propose several MILP restriction and relaxation discretizations to the pooling problem.

Finally, we report extensive computational experiments on three sets of generalized pooling problem instances, two from the literature and one introduced in this chapter. The new set of pooling problem instances being introduced here was randomly generated and are relatively harder to solve than all the previously available instances in the literature, therefore, it may serve as a new benchmark for new methodologies. Our computational results show that our technique consistently outperforms, on average, the previous methods from the literature in deriving dual bounds for pooling problem instances.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This thesis contributes to the non-convex optimization literature by generalizing important results from mixed integer linear programming to the context of mixed integer conic programming, and by introducing a number of set convexification results in the context of the quadratically constrained quadratic programming. We also illustrate the usefulness of our results with computational experiments in two applications. Some directions of future research are as follows.

Consider the convex relaxation we propose for bipartite bilinear programs (BBP) in Chapter 3. Recall that the computational success we report there depends on the sparsity of the underlying graph of the bilinear constraints defining the BBP instance. One idea that may be worth pursuing is combining set convexification with function convexification to overcome this sparsity assumption. Specifically, consider the one-constraint BBP set

$$S := \left\{ (x,y) \in [0,1]^{n_1+n_2} \;\middle|\; \sum_{(i,j)\in E} q_{ij} x_i y_j + \sum_{i \in V_1} a_i x_i + \sum_{j \in V_2} b_j y_j + c = 0 \right\},$$

where $n_1, n_2 \in \mathbb{Z}_+$, $V_1 \in \{1, \ldots, n_1\}$, $V_2 \in \{1, \ldots, n_2\}$, and $E \subseteq V_1 \times V_2$. Thus, $G = (V_1, V_2, E)$ defines the bipartite graph underlying S. The standard McCormick relaxation of S is to replace each bilinear term $w_{ij} = x_i y_j,\ (i,j) \in E$, with its convex hull, which is given by the McCormick inequalities. To obtain possibly stronger bounds, we propose the following:

1. Identify an appropriate partition of the set of edges $E = E_1 \cup E_2 \cup \cdots \cup E_p$ in the definition of S.

2. For each subset of edges, define a new variable

$$W_t = \sum_{(i,j)\in E_t} q_{ij} x_i y_j, \ \forall t \in \{1, \cdots, p\}. \tag{6.1}$$

3. Estimate bounds on W_t (for example, by minimizing and maximizing W_t over the McCormick relaxation).

4. For each $t \in \{1, \cdots, p\}$, convexify the set $S_t = \{(x, y, W_t) \in [0, 1]^{p_1^t + p_2^t + 1} : W_t = \sum_{(i,j)\in E_t} q_{ij} x_i y_j\}$, where p_1^t and p_2^t are the number of x and y variables, respectively, that are incident to E_t, and W_t has been scaled to $[0, 1]$.

An special case of the relaxation above is to define one set E_t for each edge, which becomes very similar to the McCormick relaxation. The difference is that the McCormick envelopes completely ignore bounds on W_t. Our preliminary computational experiments on a few instances from MINLPLib has shown that this approach can yield better dual bounds that those obtained with the McCormick relaxation.

Another prospective line of research has to do with the computational implications of the result of Chapter 4. Or more specifically with the proof technique used in Theorem 4.1. Part of the proof is to show that every quadratic equation can be classified into two categories. In one category, the surface defined by the quadratic equation is convex or it is the union of two convex pieces. In the other category, at each given point of the surface there exist a straight line entirely contained in the surface. We have explored this fact from a convexification point of view. However, we believe that the existence of these straight lines can have other algorithmic implications, for deriving primal bounds for example.

Finally, we were able to identify interesting combinations of linear side constraints for which we can say a lot about the convex hull of the set $\mathcal{U}^m_{(n_1,n_2)}([A^k, b_k]^m_{k=1})$ defined in (5.6). Other interesting combination of side constraint may exist, perhaps motivated by some application like the ones we obtain from the pooling problem in Chapter 5.

Appendices

APPENDIX A

OMITTED PROOFS FROM CHAPTER 5

A.1 Proof of Part (iii) of Proposition 14

Proposition 18. conv $\left(\mathcal{U}^{\text{row}}_{(n_1,n_2)}(l,u)\right)$ *is described by the inequalities (5.18), (5.19), (5.20), and (5.21).*

Proof. We will use Fourier-Motzkin elimination to obtain the convex hull in the original space. Now, we will project the t variables in the order t_{n_2}, t_{n_2-1}, t_{n_2-2}, …, t_1.

We claim that after projecting out variables $t_{n_2}, \dots, t_{n_2-j}$, the resulting system of the inequalities is:

$$
\begin{aligned}
1 - \sum_{p=1}^{n_2-(j+1)} t_p &\le \sum_{i\in\mathcal{I}} \sum_{k\in T_i} \frac{W_{ik}}{l_i} && \forall (T_1, T_2, \dots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}(\{n_2 - j, \dots, n_2\}) \\
1 - \sum_{p=1}^{n_2-(j+1)} t_p &\ge \sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} && \forall (T_1, T_2, \dots, T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2 - j, \dots, n_2\}) \\
u_{i_2} W_{i_1 k} &\ge l_{i_1} W_{i_2 k} && \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}, \forall k \in \{n_2 - j, \dots, n_2\} \\
l_i t_k \le W_{ik} &\le u_i t_k && \forall i \in [n_1], \forall k \in [n_2 - (j+1)] \\
t_k &\ge 0 && \forall k \in [n_2 - (j+1)] \\
W_{ij} &\ge 0 && \forall i \in [n_1], \forall j \in [n_2].
\end{aligned}
$$

Base case: After projecting out t_{n_2}, we obtain the system:

$$\sum_{j=1}^{n_2-1} t_j \leq 1 \tag{A.1}$$

$$W_{in_2} \leq (1 - \sum_{j=1}^{n_2-1} t_j)u_i \qquad \forall i \in [n_1] \tag{A.2}$$

$$W_{in_2} \geq (1 - \sum_{j=1}^{n_2-1} t_j)l_i \qquad \forall i \in \mathcal{I}$$

$$u_{i_2}W_{i_1n_2} \geq l_{i_1}W_{i_2n_2} \qquad \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}$$

$$l_it_j \leq W_{ij} \leq u_it_j \qquad \forall i \in [n_1], \forall j \in [n_2 - 1]$$

$$t_j \geq 0 \qquad \forall j \in [n_2 - 1],$$

$$W_{ij} \geq 0 \qquad \forall i \in [n_1], \forall j \in [n_2], \tag{A.3}$$

Note that (A.2) and (A.3) imply (A.1). Therefore the above can be written as:

$$(1 - \sum_{j=1}^{n_2-1} t_j) \leq \sum_{i\in\mathcal{I}} \sum_{k\in T_i} \frac{W_{in_2}}{l_i} \qquad \forall (T_1, T_2, \ldots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}(\{n_2\})$$

$$(1 - \sum_{j=1}^{n_2-1} t_j) \geq \sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} \qquad \forall (T_1, T_2, \ldots, T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2\})$$

$$u_{i_2}W_{i_1n_2} \geq l_{i_1}W_{i_2n_2} \qquad \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}$$

$$l_it_j \leq W_{ij} \leq u_it_j \qquad \forall i \in [n_1], \forall j \in [n_2 - 1]$$

$$t_j \geq 0 \qquad \forall j \in [n_2 - 1],$$

$$W_{ij} \geq 0 \qquad \forall i \in [n_1], \forall j \in [n_2],$$

proving the base case.

Induction step: After projecting $t_{n_2}, \ldots, t_{n_2-j}$, by the induction hypothesis we have the following system:

$$
\begin{aligned}
1 - \sum_{p=1}^{n_2-(j+2)} t_p - \sum_{i\in\mathcal{I}} \sum_{k\in T_i} \frac{W_{ik}}{l_i} &\le t_{n_2-(j+1)} && \forall (T_1, T_2, \ldots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}(\{n_2 - j, \ldots, n_2\}) \\
\frac{W_{i,n_2-(j+1)}}{u_i} &\le t_{n_2-(j+1)} && \forall i \in [n_1] \\
1 - \sum_{p=1}^{n_2-(j+2)} t_p - \sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} &\ge t_{n_2-(j+1)} && \forall (T_1, T_2, \ldots, T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2 - j, \ldots, n_2\}) \\
\frac{W_{i,n_2-(j+1)}}{l_i} &\ge t_{n_2-(j+1)} && \forall i \in \mathcal{I} \\
u_{i_2} W_{i_1 k} &\ge l_{i_1} W_{i_2 k} && \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}, \forall k \in \{n_2 - j, \ldots, n_2\} \\
l_i t_k \le W_{ik} &\le u_i t_k && \forall i \in [n_1], \forall k \in [n_2 - (j+1)] \\
t_k &\ge 0 && \forall k \in [n_2 - (j+1)] \\
W_{ik} &\ge 0 && \forall i \in [n_1], \forall k \in [n_2].
\end{aligned}
$$

Note that a constraint of the form:

$$
-\sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} \ge -\sum_{i\in\mathcal{I}} \sum_{k\in T_i'} \frac{W_{ik}}{l_i},
$$

where $(T_1, T_2 \ldots T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2-j, \ldots, n_2\})$ and $(T_1', T_2', \ldots T_{|\mathcal{I}|}') \in \mathcal{P}_{|\mathcal{I}|}(\{n_2-j, \ldots, n_2\})$ is implied by constraints of the form $u_{i_2} W_{i_1 k} \ge l_{i_1} W_{i_2 k} \forall i_2 \in [n_1], i_1 \in \mathcal{I}, k \in \{n_2 -$

$j, \ldots, n_2\}$. Thus, after projecting $t_{n_2-(j+1)}$, we obtain:

$$
\begin{aligned}
1 - \sum_{p=1}^{n_2-(j+2)} t_p &\le \sum_{i\in\mathcal{I}} \sum_{k\in T_i} \frac{W_{ik}}{l_i} && \forall (T_1, T_2, \ldots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}(\{n_2-(j+1), \ldots, n_2\}) \\
1 - \sum_{p=1}^{n_2-(j+2)} t_p &\ge \sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} && \forall (T_1, T_2, \ldots, T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2-(j+1), \ldots, n_2\}) \\
u_{i_2} W_{i_1 k} &\ge l_{i_1} W_{i_2 k} && \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}, \forall k \in \{n_2-(j+1), \ldots, n_2\} \\
l_i t_k \le W_{ik} &\le u_i t_k && \forall i \in [n_1], \forall k \in [n_2-(j+2)] \\
t_k &\ge 0 && \forall k \in [n_2-(j+2)] \\
W_{ik} &\ge 0 && \forall i \in [n_1], \forall k \in [n_2].
\end{aligned}
$$

It is straightforward now to see that after all t variables are projected, we obtain the result.

□

Proposition 19. *The inequalities in (5.18) can be separated in polynomial-time.*

Proof. For a given matrix $\hat{W}$, let us define an index j_{row} for each index $j \in [n_2]$ as

$$
j_{\text{row}} := \min\left(\operatorname*{argmax}_{i=1,\ldots,n_1}\left\{\frac{\hat{W}_{ij}}{u_i}\right\}\right). \tag{A.4}
$$

Here, we are breaking ties arbitrarily using the smallest index, when necessary. Then, we define a partition $T_1^*, \ldots, T_{n_1}^*$ of the set $[n_2]$ as

$$
T_i^* := \{j \,|\, j_{\text{row}} = i\}.
$$

Let

$$
\theta := \sum_{i=1}^{n_1} \sum_{j\in T_i^*} \frac{\hat{W}_{ij}}{u_i}.
$$

If $\theta > 1$, then a violated inequality is discovered. Otherwise, we conclude that $\hat{W}$ satisfies all the inequalities in (5.18) (by construction, the partition $T_1^*, \ldots, T_{n_1}^*$ corresponds to the inequality with the largest deviation, if one exists). Finally, we note that the complexity of this separation routine is $\mathcal{O}(n_1 n_2)$ since we need to find the maximum of n_1 numbers n_2 times to construct this partition. $\square$

Proposition 20. *The inequalities in (5.19) can be separated in polynomial-time.*

Proof. The proof is similar to the proof of Proposition 19. $\square$

A.2 Proof of Part (ii) Proposition 15

Proposition 21. conv $\left(\mathcal{U}^{\text{row+}}_{(n_1,n_2)}(l,u,L,U)\right)$ *is described by the inequalities (5.31), (5.32), (5.20), and (5.21).*

Proof. We will use Fourier-Motzkin elimination to obtain the convex hull in the original space. Now, we will project the t variables in the order $t_{n_2}, t_{n_2-1}, t_{n_2-2}, \ldots, t_1$.

We claim that after projecting out variables $t_{n_2}, \ldots, t_{n_2-j}$, the resulting system of the inequalities is:

$$
\begin{aligned}
1 - \sum_{p=1}^{n_2-(j+1)} t_p \leq \sum_{i\in\mathcal{I}} \sum_{k\in T_i} \frac{W_{ik}}{l_i} \quad & + \frac{1}{L}\sum_{i=1}^{n_1}\sum_{k\in T_0} W_{ik} \\
& \forall (T_0, T_1, \ldots, T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}(\{n_2-j, \ldots, n_2\}) \\
1 - \sum_{p=1}^{n_2-(j+1)} t_p \geq \sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} \quad & + \frac{1}{U}\sum_{i=1}^{n_1}\sum_{k\in T_0} W_{ik} \\
& \forall (T_0, T_1, \ldots, T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2-j, \ldots, n_2\}) \\
u_{i_2} W_{i_1 k} \geq l_{i_1} W_{i_2 k} \quad & \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}, \forall k \in \{n_2-j, \ldots, n_2\} \\
l_i t_k \leq W_{ik} \leq u_i t_k \quad & \forall i \in [n_1], \forall k \in [n_2-(j+1)] \\
L t_k \leq \sum_{i=1}^{n_1} W_{ik} \leq U t_k \quad & \forall k \in [n_2-(j+1)] \\
t_k \geq 0 \quad & \forall k \in [n_2-(j+1)] \\
W_{ik} \geq 0 \quad & \forall i \in [n_1], \forall k \in [n_2].
\end{aligned}
$$

Base case: After projecting out t_{n_2}, we obtain the system:

$$\sum_{j=1}^{n_2-1} t_j \leq 1 \tag{A.5}$$

$$W_{in_2} \leq (1 - \sum_{j=1}^{n_2-1} t_j) u_i \qquad \forall i \in [n_1] \tag{A.6}$$

$$W_{in_2} \geq (1 - \sum_{j=1}^{n_2-1} t_j) l_i \qquad \forall i \in \mathcal{I}$$

$$\sum_{i=1}^{n_1} W_{in_2} \leq (1 - \sum_{j=1}^{n_2-1} t_j) U$$

$$\sum_{i=1}^{n_1} W_{in_2} \geq (1 - \sum_{j=1}^{n_2-1} t_j) L$$

$$u_{i_2} W_{i_1 n_2} \geq l_{i_1} W_{i_2 n_2} \qquad \forall i_2 \in [n_1], \forall i_1 \in \mathcal{I}$$

$$l_i t_j \leq W_{ij} \leq u_i t_j \qquad \forall i \in [n_1], \forall j \in [n_2 - 1]$$

$$L t_j \leq \sum_{i=1}^{n_1} W_{ij} \leq U t_j \qquad \forall j \in [n_2 - 1]$$

$$t_j \geq 0 \qquad \forall j \in [n_2 - 1],$$

$$W_{ij} \geq 0 \qquad \forall i \in [n_1], \forall j \in [n_2], \tag{A.7}$$

Note that (A.6) and (A.7) imply (A.5). Therefore the above can be written as:

$$
\begin{aligned}
(1-\sum_{j=1}^{n_2-1} t_j) &\le \sum_{i\in\mathcal{I}}\sum_{k\in T_i}\frac{W_{in_2}}{l_i}+\frac{1}{L}\sum_{i=1}^{n_1}\sum_{k\in T_0}W_{ik} && \forall (T_0,T_1,\ldots,T_{|\mathcal{I}|})\in\mathcal{P}_{|\mathcal{I}|}(\{n_2\})\\
(1-\sum_{j=1}^{n_2-1} t_j) &\ge \sum_{i\in[n_1]}\sum_{k\in T_i}\frac{W_{ik}}{u_i}+\frac{1}{U}\sum_{i=1}^{n_1}\sum_{k\in T_0}W_{ik} && \forall (T_0,T_1,\ldots,T_{n_1})\in\mathcal{P}_{n_1}(\{n_2\})\\
u_{i_2}W_{i_1n_2} &\ge l_{i_1}W_{i_2n_2} && \forall i_2\in[n_1],\forall i_1\in\mathcal{I}\\
l_it_j \le W_{ij} &\le u_it_j && \forall i\in[n_1],\forall j\in[n_2-1]\\
Lt_j \le \sum_{i=1}^{n_1}W_{ij} &\le Ut_j && \forall j\in[n_2-1]\\
t_j &\ge 0 && \forall j\in[n_2-1],\\
W_{ij} &\ge 0 && \forall i\in[n_1],\forall j\in[n_2],
\end{aligned}
$$

proving the base case.

Induction step: After projecting t_{n_2}, ..., t_{n_2-j}, by the induction hypothesis we have the following system:

$$
\begin{aligned}
1 - \sum_{p=1}^{n_2-(j+2)} t_p - \sum_{i\in\mathcal{I}} \sum_{k\in T_i} \frac{W_{ik}}{l_i} &\leq \tfrac{1}{L}\textstyle\sum_{i=1}^{n_1}\sum_{k\in T_0} W_{ik} + t_{n_2-(j+1)} \\
&\quad \forall (T_0,\dots,T_{|\mathcal{I}|}) \in \mathcal{P}_{|\mathcal{I}|}(\{n_2-j,\dots,n_2\}) \\
\frac{W_{i,n_2-(j+1)}}{u_i} &\leq t_{n_2-(j+1)}\ \forall i\in[n_1] \\
\frac{1}{U}\sum_{i=1}^{n_1} W_{i,n_2-(j+1)} &\leq t_{n_2-(j+1)}\ \forall i\in[n_1] \\
1 - \sum_{p=1}^{n_2-(j+2)} t_p - \sum_{i\in[n_1]} \sum_{k\in T_i} \frac{W_{ik}}{u_i} &\geq \tfrac{1}{U}\textstyle\sum_{i=1}^{n_1}\sum_{k\in T_0} W_{ik} + t_{n_2-(j+1)} \\
&\quad \forall (T_0,\dots,T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2-j,\dots,n_2\}) \\
\frac{W_{i,n_2-(j+1)}}{l_i} &\geq t_{n_2-(j+1)}\ \forall i\in\mathcal{I} \\
\frac{1}{L}\sum_{i=1}^{n_1} W_{i,n_2-(j+1)} &\geq t_{n_2-(j+1)}\ \forall i\in[n_1] \\
u_{i_2}W_{i_1k} &\geq l_{i_1}W_{i_2k}\ \forall i_2\in[n_1], \forall i_1\in\mathcal{I}, \forall k\in\{n_2-j,\dots,n_2\} \\
l_it_k \leq W_{ik} &\leq u_it_k\ \forall i\in[n_1], \forall k\in[n_2-(j+1)] \\
Lt_j \leq \sum_{i=1}^{n_1} W_{ij} &\leq Ut_j\ \forall j\in[n_2-(j+1)] \\
t_k &\geq 0\ \forall k\in[n_2-(j+1)] \\
W_{ik} &\geq 0\ \forall i\in[n_1], \forall k\in[n_2].
\end{aligned}
$$

Note that a constraint of the form:

$$
-\sum_{i\in[n_1]}\sum_{k\in T_i}\frac{W_{ik}}{u_i} - \frac{1}{U}\sum_{i\in[n_1]}\sum_{k\in T_0} W_{ik} \geq -\sum_{i\in\mathcal{I}}\sum_{k\in T_i'}\frac{W_{ik}}{l_i} - \frac{1}{L}\sum_{i\in\mathcal{I}}\sum_{k\in T_0} W_{ik},
$$

where $(T_0, T_1 \dots T_{n_1}) \in \mathcal{P}_{n_1}(\{n_2-j,\dots,n_2\})$ and $(T_0', T_1', \dots T_{|\mathcal{I}|}') \in \mathcal{P}_{|\mathcal{I}|}(\{n_2-j,\dots,n_2\})$ is implied by constraints of the form $u_{i_2}W_{i_1k} \geq l_{i_1}W_{i_2k} \forall i_2 \in [n_1], i_1 \in \mathcal{I}, k \in \{n_2 -$

$j,\ldots,n_2\}$, and the fact that $L \le U$. Thus, after projecting $t_{n_2-(j+1)}$, we obtain:

$$
\begin{aligned}
1-\sum_{p=1}^{n_2-(j+2)} t_p &\le \textstyle\sum_{i\in\mathcal{I}}\sum_{k\in T_i}\frac{W_{ik}}{l_i}+\frac{1}{L}\sum_{i\in\mathcal{I}}\sum_{k\in T_0}W_{ik}\\
&\qquad \forall(T_1,T_2,\ldots,T_{|\mathcal{I}|})\in\mathcal{P}_{|\mathcal{I}|}(\{n_2-(j+1),\ldots,n_2\})\\
1-\sum_{p=1}^{n_2-(j+2)} t_p &\ge \textstyle\sum_{i\in[n_1]}\sum_{k\in T_i}\frac{W_{ik}}{u_i}+\frac{1}{U}\sum_{i\in\mathcal{I}}\sum_{k\in T_0}W_{ik}\\
&\qquad \forall(T_1,T_2,\ldots,T_{n_1})\in\mathcal{P}_{n_1}(\{n_2-(j+1),\ldots,n_2\})\\
u_{i_2}W_{i_1k} &\ge l_{i_1}W_{i_2k}\ \forall i_2\in[n_1],\forall i_1\in\mathcal{I},\forall k\in\{n_2-(j+1),\ldots,n_2\}\\
l_it_k \le W_{ik} &\le u_it_k\ \forall i\in[n_1],\forall k\in[n_2-(j+2)]\\
Lt_k \le \sum_{i=1}^{n_1}W_{ik} &\le Ut_k\ \forall\forall k\in[n_2-(j+2)]\\
t_k &\ge 0\ \forall k\in[n_2-(j+2)]\\
W_{ik} &\ge 0\ \forall i\in[n_1],\forall k\in[n_2].
\end{aligned}
$$

It is straightforward now to see that after all t variables are projected, we obtain the result.

□

Proposition 22. *The inequalities in (5.31) can be separated in polynomial-time.*

Proof. For a given matrix $\hat{W}$, let us define an index $j_{\text{row+}}$ for each index $j\in\{1,\ldots,n_2\}$ as

$$
j_{\text{row+}} := \begin{cases} 0 & \text{if } \frac{1}{U}\sum_{i=1}^{n_1}\hat{W}_{ij} \ge \max_{i=1,\ldots,n_1}\left\{\frac{\hat{W}_{ij}}{u_i}\right\} \\ j_{\text{row}} & \text{otherwise} \end{cases},
$$

where j_{row} is defined according to (A.4).

Then, we define a partition $T_0^*,T_1^*,\ldots,T_{n_1}^*$ of the set $[n_2]$ as

$$
T_i^* := \{j \mid j_{\text{row+}} = i\}.
$$

Let

$$\theta := \sum_{i=1}^{n_1} \sum_{j \in T_i^*} \frac{\hat{W}_{ij}}{u_i} + \frac{1}{U} \sum_{i=1}^{n_1} \sum_{j \in T_0^*} \hat{W}_{ij}.$$

If $\theta > 1$, then a violated inequality is discovered. Otherwise, we conclude that $\hat{W}$ satisfies all the inequalities in (5.31) (by construction, the partition $T_0^*, T_1^*, \ldots, T_m^*$ corresponds to the inequality with the largest deviation, if one exists). Finally, we note that the complexity of this separation routine is again $\mathcal{O}(n_1 n_2)$. ☐

Proposition 23. *The inequalities in (5.32) can be separated in polynomial-time.*

Proof. The proof is similar to the proof of Proposition 22. ☐

A.3 Proof of Theorem 5.4

The proof of SOC-representability follows due to Theorem 4.1 as the convex hull of a set described by the quadratic constraint $W_{11}W_{22} = W_{21}W_{12}$ and bound constraints is SOCr.

We next present an example where the convex hull of $\mathcal{U}^{\text{row}}_{(2,2)}(l,u) \cap \mathcal{U}^{\text{col}}_{(2,2)}(l,u)$ is not polyhedral.

Proposition 24. *A point of the form:*

$$\begin{bmatrix} a & \frac{a^2}{1-a} \\ 1-a & a \end{bmatrix},$$

for $a \in [0,1)$ is an extreme point of the set $\mathcal{U}^{\text{row}}_{(2,2)}(l,u) \cap \mathcal{U}^{\text{col}}_{(2,2)}(l,u)$ where $l = (0,1)$ and $u = (1,1)$.

Proof. Clearly the point is feasible. Also if it is not extreme, then it should be possible to write $(a, \frac{a^2}{1-a}) \in \mathbb{R}^2$ as a convex combination of points of the form $(a_i, \frac{a_i^2}{1-a_i}) \in \mathbb{R}^2)$ with $a_i \in [0,1) \setminus \{a\}$. However, since $f(u) = \frac{u^2}{1-u}$ is a strictly convex function, this is not possible (this is because, for example, $f(u) > \frac{a^2}{1-a} + \left(\frac{1}{(1-a)^2} - 1\right)(u-a)$ for all $u \neq a$, while $f(u) = \frac{a^2}{1-a} + \left(\frac{1}{(1-a)^2} - 1\right)(u-a)$ for $u = a$). ☐

APPENDIX B

POOLING PROBLEM INSTANCES DESCRIPTION

In Tables B.1, B.2 and B.3, AIL denotes the subset of arcs $A \cap (I \times L)$. The sets ALL, ALJ and AIJ are defined analogously. The column *Avg. Size* x^s (resp. *Avg. Size* x^t) displays the average size, over all pools $i \in L$, of the variable matrices $[x^s_{ij}]_{(s,j)}$ (resp. $[x^t_{ij}]_{(i,t)}$) for each instance.

Table B.1: Mining instances description.

Inst	$\|I\|$	$\|L\|$	$\|J\|$	$\|A\|$	$\|AIL\|$	$\|ALL\|$	$\|ALJ\|$	$\|AIJ\|$	$\|K\|$	$\|x^s\|$	Avg. Size x^s	$\|q^s\|$	$\|x^t\|$	Avg. Size x^t	$\|q^t\|$
2009H2	73	73	50	244	73	71	100	0	4	3205	(18.75, 2.34)	1369	3718	(1.97, 26.15)	1909
2009Q3	31	31	22	104	31	29	44	0	4	594	(8.26, 2.35)	256	694	(1.94, 11.90)	369
2009Q4	38	38	27	128	38	36	54	0	4	905	(10.00, 2.37)	380	1072	(1.95, 14.82)	563
2010Y	170	170	123	584	170	168	246	0	4	18038	(43.05, 2.44)	7319	21532	(1.99, 64.05)	10889
2010H1	86	86	64	298	86	84	128	0	4	4785	(22.05, 2.47)	1896	5822	(1.98, 34.59)	2975
2010H2	84	84	59	284	84	82	118	0	4	4540	(21.51, 2.38)	1807	5516	(1.98, 33.54)	2817
2010Q1	39	39	29	134	39	37	58	0	4	1068	(10.26, 2.44)	400	1356	(1.95, 18.13)	707
2010Q2	43	43	31	146	43	41	62	0	4	1196	(11.30, 2.40)	486	1444	(1.95, 17.51)	753
2010Q3	39	39	25	126	39	37	50	0	4	914	(10.26, 2.23)	400	1056	(1.95, 14.18)	553
2010Q4	43	43	32	148	43	41	64	0	4	1264	(11.26, 2.44)	484	1582	(1.95, 19.14)	823
2011Y	121	121	95	430	121	119	190	0	4	9706	(30.75, 2.55)	3721	12022	(1.98, 50.46)	6106
2011H1	67	67	50	232	67	65	100	0	4	2811	(17.25, 2.46)	1156	3344	(1.97, 25.70)	1722
2011H2	53	53	43	190	53	51	86	0	4	1993	(13.75, 2.58)	729	2548	(1.96, 24.85)	1317
2011Q1	35	35	27	122	35	33	54	0	4	784	(9.26, 2.49)	324	936	(1.94, 14.14)	495
2011Q2	30	30	22	102	30	28	44	0	4	585	(8.03, 2.40)	241	704	(1.93, 12.47)	374
2011Q3	19	19	15	66	19	17	30	0	4	260	(5.26, 2.47)	100	328	(1.89, 9.42)	179
2011Q4	28	28	23	100	28	26	46	0	4	566	(7.50, 2.57)	210	722	(1.93, 13.71)	384
2012Y	107	107	79	370	107	105	158	0	4	7394	(27.31, 2.46)	2922	9000	(1.98, 42.79)	4579
2012H1	65	65	46	220	65	63	92	0	4	2719	(16.85, 2.38)	1095	3286	(1.97, 25.98)	1689
2012H2	41	41	32	144	41	39	64	0	4	1092	(10.76, 2.51)	441	1320	(1.95, 16.88)	692
2012Q1	26	26	17	84	26	24	34	0	4	412	(7.00, 2.23)	182	478	(1.92, 9.85)	256
2012Q2	33	33	23	110	33	31	46	0	4	684	(8.76, 2.33)	289	810	(1.94, 12.97)	428
2012Q3	27	27	23	98	27	25	46	0	4	532	(7.26, 2.63)	196	680	(1.93, 13.44)	363
2012Q4	16	16	10	50	16	14	20	0	4	160	(4.50, 2.13)	72	188	(1.88, 6.50)	104

Table B.2: Literature instances description.

Inst	$\|I\|$	$\|L\|$	$\|J\|$	$\|A\|$	$\|AIL\|$	$\|ALL\|$	$\|ALJ\|$	$\|AIJ\|$	$\|K\|$	$\|x^s\|$	Avg. Size x^s	$\|q^s\|$	$\|x^t\|$	Avg. Size x^t	$\|q^t\|$
L1	3	2	3	9	3	1	3	2	1	10	(2.50, 2.00)	5	10	(2.00, 2.50)	5
L2	5	2	4	15	5	2	8	0	4	50	(5.00, 5.00)	10	28	(3.50, 4.00)	8
L3	5	2	4	15	5	2	8	0	6	50	(5.00, 5.00)	10	28	(3.50, 4.00)	8
L4	8	3	4	26	8	6	12	0	6	144	(8.00, 6.00)	24	56	(4.67, 4.00)	12
L5	8	2	5	20	8	2	10	0	4	96	(8.00, 6.00)	16	50	(5.00, 5.00)	10
L6	4	2	2	10	4	2	4	0	1	24	(4.00, 3.00)	8	12	(3.00, 2.00)	4
L12	3	2	2	9	3	2	4	0	1	18	(3.00, 3.00)	6	10	(2.50, 2.00)	4
L13	3	2	2	9	3	2	4	0	1	18	(3.00, 3.00)	6	10	(2.50, 2.00)	4
L14	3	2	2	9	3	2	4	0	1	18	(3.00, 3.00)	6	10	(2.50, 2.00)	4
L15	3	2	3	18	6	2	6	4	8	24	(3.00, 4.00)	6	24	(4.00, 3.00)	6
C2	8	6	6	71	29	9	20	13	4	194	(6.83, 4.83)	41	203	(6.33, 5.50)	33
D1	12	10	8	114	46	21	32	15	5	467	(9.10, 5.30)	91	459	(6.70, 7.00)	70

Table B.3: Random instances description.

Inst	$\|I\|$	$\|L\|$	$\|J\|$	$\|A\|$	$\|AIL\|$	$\|ALL\|$	$\|ALJ\|$	$\|AIJ\|$	$\|K\|$	$\|x^s\|$	Avg. Size x^s	$\|q^s\|$	$\|x^t\|$	Avg. size x^t	$\|q^t\|$
F1	10	10	10	84	40	12	32	0	5	241	(5.60, 4.40)	56	293	(5.20, 6.20)	62
F2	15	15	15	191	75	35	81	0	3	1030	(10.33, 7.73)	155	996	(7.33, 9.87)	148
F3	15	15	15	177	72	34	71	0	5	1150	(11.40, 7.00)	171	1268	(7.07, 12.27)	184
F4	15	15	15	187	83	31	73	0	10	1319	(13.07, 6.93)	196	1170	(7.60, 10.67)	160
F5	20	15	15	183	91	28	64	0	5	1104	(13.27, 6.13)	199	999	(7.93, 8.73)	131
F6	25	15	20	181	71	29	81	0	3	1613	(13.73, 7.33)	206	1540	(6.67, 16.73)	251
F7	20	15	25	219	82	34	103	0	8	1926	(14.53, 9.13)	218	1934	(7.73, 18.20)	273
F8	25	15	25	206	90	31	85	0	8	1617	(14.73, 7.73)	221	1543	(8.07, 13.33)	200
F9	30	10	25	179	94	12	73	0	10	1292	(16.30, 8.50)	163	1181	(10.60, 12.20)	122
F10	25	15	30	360	72	19	111	158	8	1155	(10.40, 8.67)	156	1013	(6.07, 12.73)	191
F11	10	10	10	103	32	15	26	30	5	230	(6.20, 4.10)	62	220	(4.70, 5.20)	52
F12	15	15	15	194	60	26	55	53	3	697	(8.80, 5.40)	132	839	(5.73, 10.60)	159
F13	15	15	15	247	71	32	75	69	5	1196	(11.80, 7.13)	177	1048	(6.87, 10.40)	156
F14	15	15	15	220	61	29	69	61	10	841	(9.47, 6.53)	142	835	(6.00, 10.33)	155
F15	20	15	15	244	80	33	59	72	5	1147	(13.47, 6.13)	202	1066	(7.53, 10.13)	152
F16	25	15	20	188	59	9	45	75	3	445	(7.60, 3.60)	114	402	(4.53, 5.33)	80
F17	20	15	25	167	40	15	39	73	8	313	(6.20, 3.60)	93	343	(3.67, 6.27)	94
F18	25	15	25	201	47	17	61	76	8	609	(8.27, 5.20)	124	557	(4.27, 9.27)	139
F19	30	10	25	186	41	3	33	109	10	223	(6.10, 3.60)	61	212	(4.40, 4.50)	45
F20	30	20	25	220	58	24	59	79	5	509	(7.65, 4.15)	153	490	(4.10, 6.85)	137
F21	30	35	25	566	241	117	208	0	5	7618	(24.31, 9.29)	851	6787	(10.23, 19.77)	692
F22	35	15	40	347	66	12	85	184	8	786	(8.33, 6.47)	125	781	(5.20, 9.53)	143
F23	35	40	35	536	215	119	202	0	5	6634	(21.93, 8.03)	877	6274	(8.35, 20.93)	837
F24	25	15	30	355	78	26	78	173	8	1338	(13.87, 6.93)	208	1330	(6.93, 13.93)	209

REFERENCES

[1] W. Adams, A. Gupte, and Y. Xu, "Error bounds for monomial convexification in polynomial optimization," *Mathematical Programming*, 2018.

[2] N. Adhya, M. Tawarmalani, and N. V. Sahinidis, "A Lagrangian approach to the pooling problem," *Industrial & Engineering Chemistry Research*, vol. 38, no. 5, pp. 1956–1972, 1999.

[3] A. A. Ahmadi and A. Majumdar, "Dsos and sdsos optimization: Lp and socp-based alternatives to sum of squares optimization," in *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, IEEE, 2014, pp. 1–5.

[4] F. A. Al-Khayyal and J. E. Falk, "Jointly constrained biconvex programming," *Mathematics of Operations Research*, vol. 8, no. 2, pp. 273–286, 1983.

[5] M. Alfaki and D. Haugland, "Strong formulations for the pooling problem," *Journal of Global Optimization*, vol. 56, no. 3, pp. 897–916, 2013.

[6] M. Alfaki and D. Haugland, "A multi-commodity flow formulation for the generalized pooling problem," *Journal of Global Optimization*, vol. 56, no. 3, pp. 917–937, 2013.

[7] E. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.

[8] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, "Alphabb: A global optimization method for general constrained nonconvex problems," *Journal of Global Optimization*, vol. 7, no. 4, pp. 337–363, 1995.

[9] K. M. Anstreicher and S. Burer, "Computable representations for convex hulls of low-dimensional quadratic forms," *Mathematical programming*, vol. 124, no. 1-2, pp. 33–43, 2010.

[10] M. ApS, *The mosek optimization toolbox for matlab manual. version 8.1.* 2017.

[11] A. Atamtürk and V. Narayanan, "Conic mixed-integer rounding cuts," *Mathematical Programming*, vol. 122, no. 1, pp. 1–20, 2010.

[12] Y. Q. Bai and G. Q. Wang, “Primal-dual interior-point algorithms for second-order cone optimization based on a new parametric kernel function,” *Acta Mathematica Sinica, English Series*, vol. 23, no. 11, pp. 2027–2042, 2007.

[13] R. Baltean-Lugojan and R. Misener, “Piecewise parametric structure in the pooling problem: From sparse strongly-polynomial solutions to NP-hardness,” *Journal of Global Optimization*, vol. 71, no. 4, pp. 655–690, 2018.

[14] X. Bao, A. Khajavirad, N. V. Sahinidis, and M. Tawarmalani, “Global optimization of nonconvex problems with multilinear intermediates,” *Mathematical Programming Computation*, vol. 7, no. 1, pp. 1–37, 2015.

[15] A. Basu, M. Conforti, G. Cornuéjols, and G. Zambelli, “Maximal lattice-free convex sets in linear subspaces,” *Mathematics of Operations Research*, vol. 35, no. 3, pp. 704–720, 2010.

[16] A. Basu, R. Hildebrand, and M. Köppe, “Light on the infinite group relaxation i: Foundations and taxonomy,” *4OR*, vol. 14, no. 1, pp. 1–40, 2016.

[17] P. Belotti, J. C. Góez, I. Pólik, T. K. Ralphs, and T. Terlaky, “On families of quadratic surfaces having fixed intersections with two hyperplanes,” *Discrete Applied Mathematics*, vol. 161, no. 16-17, pp. 2778–2793, 2013.

[18] P. Belotti, A. J. Miller, and M. Namazifar, “Valid inequalities and convex hulls for multilinear functions,” *Electronic Notes in Discrete Mathematics*, vol. 36, pp. 805–812, 2010.

[19] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001. eprint: `http://epubs.siam.org/doi/pdf/10.1137/1.9780898718829`.

[20] H. P. Benson, “Concave envelopes of monomial functions over rectangles,” *Naval Research Logistics (NRL)*, vol. 51, no. 4, pp. 467–476, 2004.

[21] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st. Athena Scientific, 1997, ISBN: 1886529191.

[22] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman, “Julia: A fast dynamic language for technical computing,” *CoRR*, vol. abs/1209.5145, 2012. arXiv: `1209.5145`.

[23] D. Bienstock, C. Chen, and G. Munoz, “Outer-product-free sets for polynomial optimization and oracle-based cuts,” *arXiv preprint arXiv:1610.04604*, 2016.

[24] C. E. Blair and R. G. Jeroslow, "The value function of an integer program," *Mathematical Programming*, vol. 23, no. 1, pp. 237–273, 1982.

[25] M. Bodur, A. Del Pia, S. S. Dey, M. Molinaro, and S. Pokutta, "Aggregation-based cutting-planes for packing and covering integer programs," *Mathematical Programming*, 2017.

[26] N. Boland, T. Kalinowski, and F. Rigterink, "A polynomially solvable case of the pooling problem," *Journal of Global Optimization*, vol. 67, no. 3, pp. 621–630, 2017.

[27] N. Boland, S. S. Dey, T. Kalinowski, M. Molinaro, and F. Rigterink, "Bounding the gap between the mccormick relaxation and the convex hull for bilinear functions," *Math. Program.*, vol. 162, no. 1-2, pp. 523–535, 2017.

[28] N. Boland, T. Kalinowski, and F. Rigterink, "New multi-commodity flow formulations for the pooling problem," *Journal of Global Optimization*, vol. 66, no. 4, pp. 669–710, 2016.

[29] N. Boland, T. Kalinowski, F. Rigterink, and M. Savelsbergh, "A special case of the generalized pooling problem arising in the mining industry," *Preprint at `http://www.optimization-online.org/DB_HTML/2015/07/5025.html`*, 2015.

[30] P. Bonami, O. Günlük, and J. Linderoth, "Globally solving nonconvex quadratic programming problems with box constraints via integer programming methods," *Mathematical Programming Computation*, vol. 10, no. 3, pp. 333–382, 2018.

[31] S. Bose, D. F. Gayme, K. M. Chandy, and S. H. Low, "Quadratically constrained quadratic programs on acyclic graphs with application to power flow," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 278–287, 2015.

[32] J. J. M. Bront, I. Mndez-Daz, and G. Vulcano, "A column generation algorithm for choice-based network revenue management," *Operations Research*, vol. 57, no. 3, pp. 769–784, 2009. eprint: `https://doi.org/10.1287/opre.1080.0567`.

[33] C. Buchheim and C. D'Ambrosio, "Monomial-wise optimal separable underestimators for mixed-integer polynomial optimization," *Journal of Global Optimization*, vol. 67, no. 4, pp. 759–786, 2017.

[34] S. Burer and F. Kılınç-Karzan, "How to convexify the intersection of a second order cone and a nonconvex quadratic," *Mathematical Programming*, vol. 162, no. 1, pp. 393–429, 2017.

[35] S. Burer, S. Kim, and M. Kojima, "Faster, but weaker, relaxations for quadratically constrained quadratic programs," *Computational Optimization and Applications*, vol. 59, no. 1, pp. 27–45, 2014.

[36] S. Burer and A. N. Letchford, "On nonconvex quadratic programming with box constraints," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 1073–1089, 2009. eprint: `https://doi.org/10.1137/080729529`.

[37] S. Burer and R. D. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[38] P. M. Castro, "Tightening piecewise mccormick relaxations for bilinear problems," *Computers & Chemical Engineering*, vol. 72, pp. 300–311, 2015.

[39] M. Çezik and G. Iyengar, "Cuts for mixed 0-1 conic programming," *Mathematical Programming*, vol. 104, no. 1, pp. 179–202, 2005.

[40] M. Chandraker and D. Kriegman, "Globally optimal bilinear programming for computer vision applications," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[41] C. Chen, A. Atamtrk, and S. S. Oren, "Bound tightening for the alternating current optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3729–3736, 2016.

[42] M. Conforti, G. Cornuejols, and G. Zambelli, *Integer Programming*, ser. Graduate Texts in Mathematics. Springer International Publishing, 2014, ISBN: 978-3-319-38432-0.

[43] M. Conforti, G. Cornujols, A. Daniilidis, C. Lemarchal, and J. Malick, "Cut-generating functions and s-free sets," *Mathematics of Operations Research*, vol. 40, no. 2, pp. 276–391, 2015. eprint: `https://doi.org/10.1287/moor.2014.0670`.

[44] Y. Crama and E. Rodríguez-Heck, "A class of valid inequalities for multilinear 0–1 optimization problems," *Discrete Optimization*, vol. 25, pp. 28–47, 2017.

[45] H. Crowder, E. L. Johnson, and M. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research*, vol. 31, no. 5, pp. 803–834, 1983.

[46] D. Dadush, S. S. Dey, and J. P. Vielma, "The split closure of a strictly convex body," *Operations Research Letters*, vol. 39, no. 2, pp. 121–126, 2011.

[47] S. Dash and O. Günlük, "Valid inequalities based on simple mixed integer set," *Mathematical Programming*, vol. 106, pp. 29–53, 2006.

[48] D. Davarnia, J.-P. P. Richard, and M. Tawarmalani, "Simultaneous convexification of bilinear functions over polytopes with application to network interdiction," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1801–1833, 2017.

[49] A. Del Pia and A. Khajavirad, "A polyhedral study of binary polynomial programs," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 389–410, 2016.

[50] S. S. Dey and J.-P. P. Richard, "Facets of the two-dimensional infinite group problems," *Mathematics of Operations Research*, vol. 33, pp. 140–166, 2008.

[51] S. S. Dey and A. Gupte, "Analysis of MILP techniques for the pooling problem," *Operations Research*, vol. 63, no. 2, pp. 412–427, 2015. eprint: `https://doi.org/10.1287/opre.2015.1357`.

[52] S. S. Dey, M. Molinaro, and Q. Wang, "Analysis of sparse cutting planes for sparse milps with applications to stochastic milps," *Mathematics of Operations Research*, 2017.

[53] S. S. Dey and D. A. Morán R., "Some properties of convex hulls of integer points contained in general convex sets," *Mathematical Programming*, vol. 141, no. 1, pp. 507–526, 2013.

[54] S. S. Dey and J.-P. P. Richard, "Some relations between facets of low- and high-dimensional group problems," *Mathematical Programming*, vol. 123, pp. 285–313, 2010.

[55] S. S. Dey, A. Santana, and Y. Wang, "New socp relaxation and branching rule for bipartite bilinear programs," *Optimization and Engineering*, 2018.

[56] S. S. Dey and L. A. Wolsey, "Two row mixed-integer cuts via lifting," *Mathematical Programming*, vol. 124, no. 1, pp. 143–174, 2010.

[57] W. L. Edge, *The Theory of Ruled Surfaces*. Cambridge University Press, 2011, ISBN: 9781107689671.

[58] D. C. Faria and M. J. Bagajewicz, "Novel bound contraction procedure for global optimization of bilinear minlp problems with applications to water management problems," *Computers & chemical engineering*, vol. 35, no. 3, pp. 446–455, 2011.

[59] H. Fawzi, "On representing the positive semidefinite cone using the second-order cone," *Mathematical Programming*, pp. 1–10, 2018.

[60] B. Galan and I. E. Grossmann, "Optimal design of distributed wastewater treatment networks," *Industrial & Engineering Chemistry Research*, vol. 37, no. 10, pp. 4036–4048, 1998. eprint: `http://dx.doi.org/10.1021/ie980133h`.

[61] A. Gharanjik, B. Shankar, M. Soltanalian, and B. Oftersten, "An iterative approach to nonconvex qcqp with applications in signal processing," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016 IEEE*, IEEE, 2016, pp. 1–5.

[62] M. X. Goemans and D. P. Williamson, ". 879-approximation algorithms for max cut and max 2sat," in *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, ACM, 1994, pp. 422–431.

[63] R. E. Gomory and E. L. Johnson, "Some continuous functions related to corner polyhedra, part I," *Mathematical Programming*, vol. 3, pp. 23–85, 1972.

[64] R. E. Gomory and E. L. Johnson, "T-space and cutting planes," *Mathematical Programming*, vol. 96, pp. 341–375, 2003.

[65] J. Gorski, F. Pfeuffer, and K. Klamroth, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.

[66] A. Gupte, "Mixed integer bilinear programming with applications to the pooling problem," PhD thesis, Georgia Institute of Technology, 2011.

[67] A. Gupte, S. Ahmed, S. S. Dey, and M. S. Cheon, "Relaxations and discretizations for the pooling problem," *Journal of Global Optimization*, vol. 67, no. 3, pp. 631–669, 2017.

[68] A. Gupte, T. Kalinowski, F. Rigterink, and H. Waterer, "Extended formulations for convex hulls of graphs of bilinear functions," *arXiv preprint arXiv:1702.04813*, 2017.

[69] D. Haugland, "The computational complexity of the pooling problem," *Journal of Global Optimization*, vol. 64, no. 2, pp. 199–215, 2016.

[70] D. Haugland and E. M. Hendrix, "Pooling problems with polynomial-time algorithms," *Journal of Optimization Theory and Applications*, vol. 170, no. 2, pp. 591–615, 2016.

[71] C. A. Haverly, "Studies of the behavior of recursion for the pooling problem," *Acm sigmap bulletin*, no. 25, pp. 19–28, 1978.

[72] R. J. Hillestad and S. E. Jacobsen, "Linear programs with an additional reverse convex constraint," *Applied Mathematics and Optimization*, vol. 6, no. 1, pp. 257–269, 1980.

[73] A. J. Hoffman, "Binding constraints and helly numbers," *Annals of the New York Academy of Sciences*, vol. 319, no. 1, pp. 284–288, 1979.

[74] R. G. Jeroslow, "Minimal inequalities," *Mathematical Programming*, vol. 17, no. 1, pp. 1–15, 1979.

[75] E. L. Johnson, "Cyclic groups, cutting planes, shortest paths," in *Mathematical Programming*, T. Hu and S. M. Robinson, Eds., Academic Press, 1973, pp. 185 –211, ISBN: 978-0-12-358350-5.

[76] E. L. Johnson, "On the group problem and a subadditive approach to integer programming," in *Discrete Optimization II*, ser. Annals of Discrete Mathematics, P. Hammer, E. Johnson, and B. Korte, Eds., vol. 5, Elsevier, 1979, pp. 97 –112.

[77] A. Khabbazibasmenj and S. A. Vorobyov, "Generalized quadratically constrained quadratic programming for signal processing," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7629–7633.

[78] F. Kılınç-Karzan and D. E. Steffy, "On sublinear inequalities for mixed integer conic programs," *Mathematical Programming*, vol. 159, no. 1, pp. 585–605, 2016.

[79] F. Kilinç-Karzan and S. Yildiz, "Two-term disjunctions on the second-order cone," *Mathematical Programming*, vol. 154, no. 1-2, pp. 463–491, 2015.

[80] F. Kılınç-Karzan, "On minimal valid inequalities for mixed integer conic programs," *Mathematics of Operations Research*, vol. 41, no. 2, pp. 477–510, 2016.

[81] S. Kim and M. Kojima, "Second order cone programming relaxation of nonconvex quadratic optimization problems," *Optimization methods and software*, vol. 15, no. 3-4, pp. 201–224, 2001.

[82] B. Kocuk, S. S. Dey, and X. A. Sun, "Matrix minor reformulation and SOCP-based spatial branch-and-cut method for the AC optimal power flow problem," *Mathematical Programming Computation*, vol. 10, no. 4, pp. 557–596, 2018.

[83] B. Kocuk, S. S. Dey, and X. A. Sun, "Strong socp relaxations for the optimal power flow problem," *Operations Research*, vol. 64, no. 6, pp. 1177–1196, 2016.

[84] H. Konno and R. Yamamoto, "Global optimization versus integer programming in portfolio optimization under nonconvex transaction costs," *Journal of Global Optimization*, vol. 32, no. 2, pp. 207–219, 2005.

[85] M. Köppe and Y. Zhou, "An electronic compendium of extreme functions for the gomory–johnson infinite group problem," *arXiv:1411.5121*, 2015.

[86] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[87] Q. Li and V. Vittal, "Convex hull of the quadratic branch ac power flow equations and its application in radial distribution networks," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 839–850, 2018.

[88] L. Liberti and C. C. Pantelides, "Convex envelopes of monomials of odd degree," *Journal of Global Optimization*, vol. 25, no. 2, pp. 157–168, 2003.

[89] J. Linderoth, "A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs," *Mathematical Programming*, vol. 103, no. 2, pp. 251–282, 2005.

[90] M. Locatelli, "Polyhedral subdivisions and functional forms for the convex envelopes of bilinear, fractional and other bivariate functions over general polytopes," *Journal of Global Optimization*, vol. 66, no. 4, pp. 629–668, 2016.

[91] J. Luedtke, C. D'Ambrosio, J. Linderoth, and J. Schweiger, "Strong convex nonlinear relaxations of the pooling problem," *arXiv preprint arXiv:1803.02955*, 2018.

[92] J. R. Luedtke, M. Namazifar, and J. Linderoth, "Some results on the strength of relaxations of multilinear functions," *Math. Program.*, vol. 136, no. 2, pp. 325–351, 2012.

[93] A. Makrodimopoulos, A. Bhaskar, and A. J. Keane, "Second-order cone programming formulations for a class of problems in structural optimization," *Structural and Multidisciplinary Optimization*, vol. 40, no. 1, p. 365, 2009.

[94] H. Marchand and L. A. Wolsey, "Aggregation and mixed integer rounding to solve mips," *Operations research*, vol. 49, no. 3, pp. 363–371, 2001.

[95] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part iconvex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[96] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM Journal on Optimization*, vol. 2, no. 4, pp. 575–601, 1992. eprint: `https://doi.org/10.1137/0802028`.

[97] C. A. Meyer and C. A. Floudas, “Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes,” *Journal of Global Optimization*, vol. 29, no. 2, pp. 125–155, 2004.

[98] C. A. Meyer and C. A. Floudas, “Global optimization of a combinatorially complex generalized pooling problem,” *AIChE journal*, vol. 52, no. 3, pp. 1027–1037, 2006.

[99] C. A. Meyer and C. A. Floudas, “Convex envelopes for edge-concave functions,” *Mathematical programming*, vol. 103, no. 2, pp. 207–224, 2005.

[100] R. Misener, J. B. Smadbeck, and C. A. Floudas, “Dynamically generated cutting planes for mixed-integer quadratically constrained quadratic programs and their incorporation into glomiqo 2,” *Optimization Methods and Software*, vol. 30, no. 1, pp. 215–249, 2015.

[101] R. Misener, J. P. Thompson, and C. A. Floudas, “Apogee: Global optimization of standard, generalized, and extended pooling problems via linear and logarithmic partitioning schemes,” *Computers and Chemical Engineering*, vol. 35, no. 5, pp. 876 –892, 2011, Selected Papers from ESCAPE-20 (European Symposium of Computer Aided Process Engineering - 20), 6-9 June 2010, Ischia, Italy.

[102] S. Modaresi, M. R. Kilinç, and J. P. Vielma, “Split cuts and extended formulations for mixed integer conic quadratic programming,” *Operations Research Letters*, vol. 43, no. 1, pp. 10–15, 2015.

[103] S. Modaresi and J. P. Vielma, “Convex hull of two quadratic or a conic quadratic and a quadratic inequality,” *http://www.optimization-online.org/DBHTML/2014/11/4641.html*, 2014.

[104] S. Modaresi and J. P. Vielma, “Convex hull of two quadratic or a conic quadratic and a quadratic inequality,” *Mathematical Programming*, vol. 164, no. 1-2, pp. 383–409, 2017.

[105] D. A. Morn, S. S. Dey, and J. P. Vielma, “A strong dual for conic mixed-integer programs,” *SIAM Journal on Optimization*, vol. 22, no. 3, pp. 1136–1150, 2012. eprint: `https://doi.org/10.1137/110840868`.

[106] L. Mustonen, X. Gao, A. Santana, R. Mitchell, Y. Vigfusson, and L. Ruthotto, “A bayesian framework for molecular strain identification from mixed diagnostic samples,” *Inverse Problems*, vol. 34, no. 10, p. 105 009, 2018.

[107] A. G. Nahapetyan, “Bilinear programming: Applications in the supply chain management bilinear programming: Applications in the supply chain management,” in *Encyclopedia of Optimization*, Springer, 2008, pp. 282–288.

[108] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, ser. Wiley-Interscience series in discrete mathematics and optimization. New York: Wiley, 1988, Edition broche : 1999, ISBN: 0-471-82819-X.

[109] G. L. Nemhauser and L. A. Wolsey, "A recursive procedure to generate all cuts for 0–1 mixed integer programs," *Mathematical Programming*, vol. 46, no. 1, pp. 379–390, 1990.

[110] T. T. Nguyen, J.-P. P. Richard, and M. Tawarmalani, "Deriving the convex hull of a polynomial partitioning set through lifting and projection," working paper, Tech. Rep., 2013.

[111] T. T. Nguyen, M. Tawarmalani, and J.-P. P. Richard, "Convexification techniques for linear complementarity constraints.," in *IPCO*, Springer, vol. 6655, 2011, pp. 336–348.

[112] J. Nocedal and S. Wright, *Numerical Optimization*. Springer-Verlag New York, 2006.

[113] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, 2016.

[114] M. Padberg, "The boolean quadric polytope: Some characteristics, facets and relatives," *Mathematical programming*, vol. 45, no. 1-3, pp. 139–172, 1989.

[115] H. Rahman and A. Mahajan, "Facets of a mixed-integer bilinear covering set with bounds on variables," *arXiv preprint arXiv:1707.06712*, 2017.

[116] J. P. Richard and S. S. Dey, "The group-theoretic approach in mixed integer programming," in *50 Years of Integer Programming 1958-2008 - From the Early Years to the State-of-the-Art*, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds., Springer, 2010, pp. 727–801, ISBN: 978-3-540-68274-5.

[117] J. P. Richard, Y. Li, and L. A. Miller, "Valid inequalities for mips and group polyhedra from approximate liftings," *Mathematical Programming*, vol. 118, no. 2, pp. 253–277, 2009.

[118] A. D. Rikun, "A convex envelope formula for multilinear functions," *Journal of Global Optimization*, vol. 10, no. 4, pp. 425–437, 1997.

[119] H. S. Ryoo and N. V. Sahinidis, "A branch-and-reduce approach to global optimization," *Journal of Global Optimization*, vol. 8, no. 2, pp. 107–138, 1996.

[120] H. S. Ryoo and N. V. Sahinidis, “Analysis of bounds for multilinear functions,” *Journal of Global Optimization*, vol. 19, no. 4, pp. 403–424, 2001.

[121] N. V. Sahinidis, “Baron: A general purpose global optimization software package,” *Journal of Global Optimization*, vol. 8, no. 2, pp. 201–205, 1996.

[122] N. V. Sahinidis and M. Tawarmalani, “Accelerating branch-and-bound through a modeling language construct for relaxation-specific constraints,” *Journal of Global Optimization*, vol. 32, no. 2, pp. 259–280, 2005.

[123] A. Santana and S. S. Dey, “Some cut-generating functions for second-order conic sets,” *Discrete Optimization*, vol. 24, pp. 51 –65, 2017, Conic Discrete Optimization.

[124] A. Schrijver, *Theory of linear and integer programming*, ser. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999, ISBN: 978-0-471-98232-6.

[125] H. D. Sherali and A. Alameddine, “An explicit characterization of the convex envelope of a bivariate bilinear function over special polytopes,” *Annals of Operations Research*, vol. 25, no. 1, pp. 197–209, 1990.

[126] H. D. Sherali, “Convex envelopes of multilinear functions over a unit hypercube and over special discrete sets,” *Acta mathematica vietnamica*, vol. 22, no. 1, pp. 245–270, 1997.

[127] H. D. Sherali and W. P. Adams, *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Springer Science & Business Media, 2013, vol. 31.

[128] H. D. Sherali and A. Alameddine, “A new reformulation-linearization technique for bilinear programming problems,” *Journal of Global optimization*, vol. 2, no. 4, pp. 379–410, 1992.

[129] A. Skajaa, E. D. Andersen, and Y. Ye, “Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems,” *Mathematical Programming Computation*, vol. 5, no. 1, pp. 1–25, 2013.

[130] E. Speakman and J. Lee, “On branching-point selection for triple products in spatial branch-and-bound: The hull relaxation,” *arXiv preprint arXiv:1706.08438*, 2017.

[131] E. Speakman and J. Lee, “Quantifying double mccormick,” *Mathematics of Operations Research*, vol. 42, no. 4, pp. 1230–1253, 2017.

[132] K. Talluri and G. van Ryzin, “Revenue management under a general discrete choice model of consumer behavior,” *Management Science*, vol. 50, no. 1, pp. 15–33, 2004. eprint: `https://doi.org/10.1287/mnsc.1030.0147`.

[133] M. Tawarmalani and J.-P. P. Richard, “Decomposition techniques in convexification of inequalities,” *Technical report*, 2013.

[134] M. Tawarmalani, J.-P. P. Richard, and K. Chung, “Strong valid inequalities for orthogonal disjunctions and bilinear covering sets,” *Mathematical Programming*, vol. 124, no. 1, pp. 481–512, 2010.

[135] M. Tawarmalani, J.-P. P. Richard, and C. Xiong, “Explicit convex and concave envelopes through polyhedral subdivisions,” *Mathematical Programming*, pp. 1–47, 2013.

[136] M. Tawarmalani and N. V. Sahinidis, *Convexification and global optimization in continuous and mixed-integer nonlinear programming: theory, algorithms, software, and applications*. Springer Science & Business Media, 2002, vol. 65.

[137] H. Tuy, *Convex analysis and global optimization*. Springer, 2016, vol. 110.

[138] G. Wang and Y. Bai, “A primal-dual interior-point algorithm for second-order cone optimization with full nesterovtodd step,” *Applied Mathematics and Computation*, vol. 215, no. 3, pp. 1047 –1061, 2009.

[139] L. Wolsey, “The b-hull of an integer program,” *Discrete Applied Mathematics*, vol. 3, no. 3, pp. 193 –201, 1981.

[140] S. Yildiz and G. Cornuéjols, “Disjunctive cuts for cross-sections of the second-order cone,” *Operations Research Letter*, vol. 43, no. 4, pp. 432–437, 2015.

[141] D. Zhu, X. Dong, and Y. Wang, “Substructure stiffness and mass updating through minimization of modal dynamic residuals,” *Journal of Engineering Mechanics*, vol. 142, no. 5, p. 04 016 013, 2016.